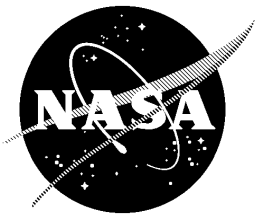
514-3SCG/0197

MISSION OPERATIONS AND DATA SYSTEMS DIRECTORATE

Landsat 7 Processing System (LPS) Software Configuration Guide

August 1997



National Aeronautics and Space Administration

Goddard Space Flight Center
Greenbelt, Maryland

Landsat 7 Processing System (LPS) Software Configuration Guide

August 1997

Prepared by:

Charlene Binley — Date
Software Configuration Manager

Quality Assured by:

Sheila Whisonant — Date
Quality Assurance Officer

Concurred by:

Khemraj Sharma — Date
Integration and Test Department Deputy Manager

Approved by:

Nate Daniel — Date
LPS Element Manager

Joy Henegar — Date
LPS Project Manager

Goddard Space Flight Center
Greenbelt, Maryland

Abstract

This document describes the allocation of software and data to physical disks on the development and operational Landsat 7 Processing System (LPS). Furthermore, it describes the software configuration management being applied to LPS, which includes procedural information for installing LPS and for building the LPS executables and installing them for testing and operations.

Contents

1. Introduction

2. Physical and Logical Disk Assignments

3. Configuration Management

List of Tables

List of Figures

Section 1. Introduction

1.1 Purpose

The LPS Software Configuration Guide is intended to provide in-depth procedural information for applying configuration management (CM) to the Landsat 7 Processing System (LPS) application software located at the customer site. This guide also explains how to build the LPS executables and where the new executables can be tested for verification.

Applicable documentation for this guide is listed in the References section.

1.2 Environments

LPS consists of five individual computers with identical operational environments. Each computer and its operational environment defines a string. Four of the strings support normal operations at all times. The fifth string is available for LPS test and maintenance support, as required, and as a backup string to the four operational strings. References 1 through 3 provide more specific information on the five LPS strings.

LPS provides a development environment on the fifth string for maintenance programmers who perform software enhancements and software corrections.

LPS also provides a test environment on the fifth string to test and verify changes to the LPS software before they are propagated to the four operational strings. The purpose and function of the test environment is further described in Reference 1.

LPS provides a CM environment on the fifth string to control access to the latest LPS operational software. LPS uses the Polytron Version Control System (PVCS) Version Manager as its CM tool.

1.3 Directory Structure for the Operational Environment

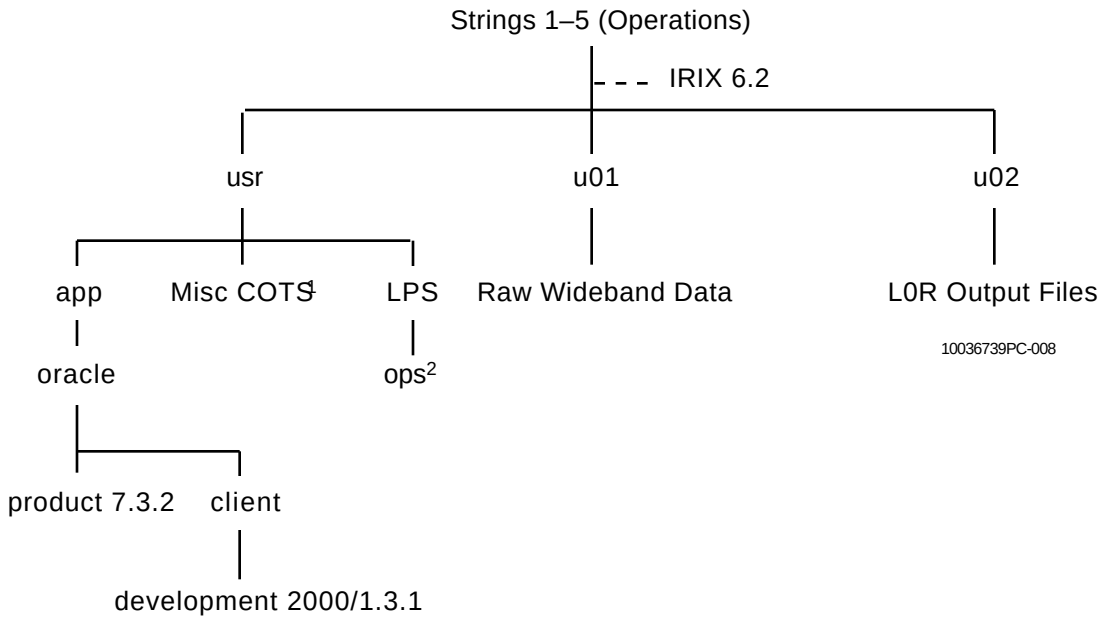
The directory structure of the operational environment (Figure 1–1) is defined in Reference 1. The four operational strings contain executable code and do not contain any application source files. New versions of the executables are generated and tested on the backup string and, if approved, promoted to the operational environment on the four LPS strings, as well as the operational directory on the fifth string.

1.4 Directory Structure for the Test Environment

The directory structure of the test environment is reflected in Figure 1–2.

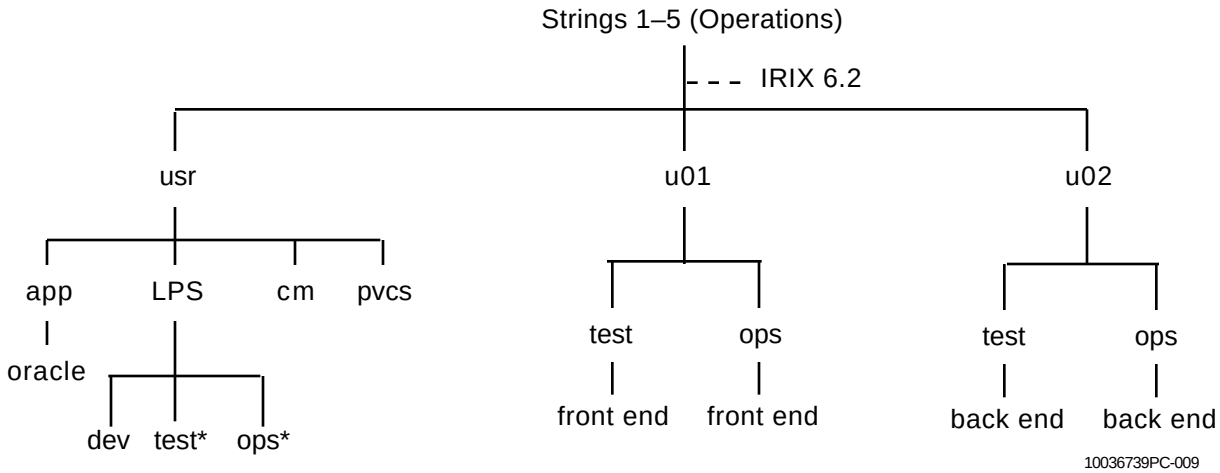
1.5 Directory Structure for the Development Environment

The directory structure of the development environment on string 5 stores developed software as shown in Figure 1–2. Source code (Rs) is stored in src and includes subdirectories; the executables (Rx) for the LPS applications are stored in the bin directory.



[1]Installation of the operating system, IRIX 6.2, automatically populates the miscellaneous COT
[2]Current version of the executables; see Figure 3–4 for the operational string structure.

Figure 1–1. Operational Environment Directory Structure

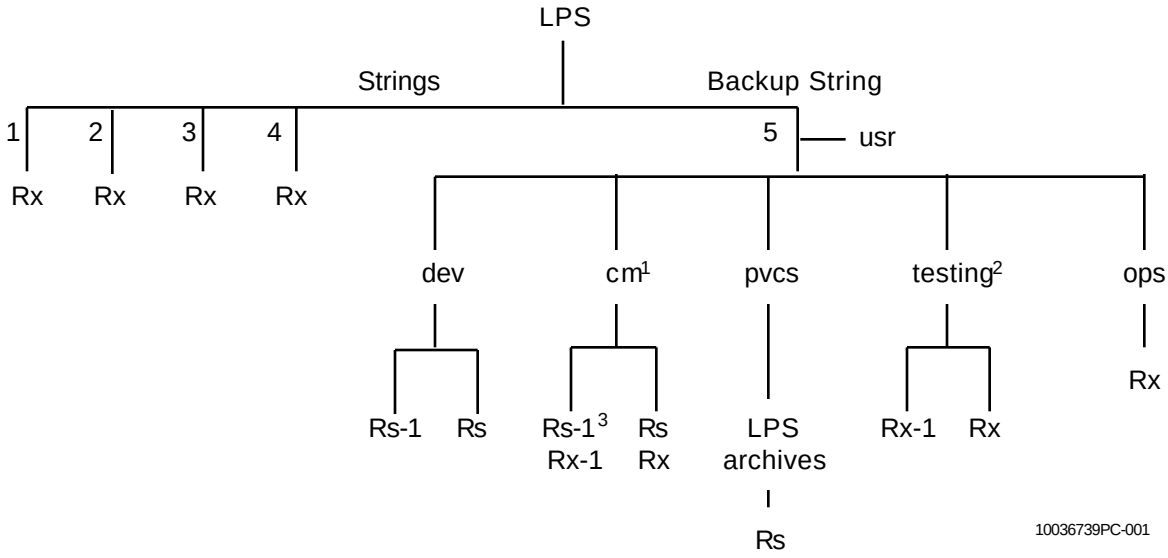


*See Figures 3–3 and 3–4 for more detail.

Figure 1–2. String 5 – Test Environment Directory Structure

1.5.1 Directory Structure for Source and Executables on Development String

The basic structure for storing developed software on string 5 is shown in Figure 1–3. Figures 3–2 and 3–3 show more detail of the directory structure. The "pvcs" subdirectory stores source code, and the "cm" subdirectory stores CM-related scripts and build LPS executables. The "ops" subdirectory houses LPS executables. Separate subdirectories for each build permit quick access to the current and previous source code. Once accepted, the LPS executable files are copied to the appropriate subdirectory on strings 1 to 4. Thus, the executables will be stored on all five strings.



Rs = current release's source for that environment

Rs-1 = previous release's source for that environment

Rx = current release's executables for that environment

Rx-1 = previous release's executables for that environment

[1]See Figure 3–1

[2]See Figure 3–2

[3]See Figure 3–3

Figure 1–3. Basic Structure for Storing Developed Software

Section 2. Physical and Logical Disk Assignments

This section describes the physical disks on the operational and backup string disks and their logical allocation for the development, test, and operational environments. Appendix A provides a checklist for installing the required LPS-related software.

2.1 Operational String Disks

Each operational string has a system disk and two Redundant Array of Inexpensive Devices (RAIDs), one for incoming data from the Landsat Ground Station (LGS), and a second RAID for temporary storage of LPS output data sets awaiting retrieval by the EOS Core System.

2.1.1 Logical-to-Physical Disk Mapping of Operational String

This section describes the logical disk allocation defined for the physical disks. Table 2–1 summarizes the allocation of the logical file systems to physical disks. This allocation is expected to meet operational needs.

Table 2–1. Summary of Disk Allocation for Operational String

Disk	File System	Use	Allocation
System	/ /usr /usr/LPS/ops	Operations	IRIX 6.2 ORACLE (Server and ORACLE Developer 2000 1.3.1) Other commercial off-the-shelf (COTS) software (Section 2.1.1.1) LPS executables
RAID 1	/u01	Front end	Raw wideband data
RAID 2	/u02	Back end	L0R output files

2.1.1.1 System Disk in Each Operational String

At any one time, the system disk on each operational string will contain the following:

- Operating system and associated files
- One operational instance of the LPS database, one for each environment (development, test, and operational)
- COTS
  - IRIX 6.2 (operating system)
  - ONC3/NFS 6.2
  - IRIS Development Option 6.2 (CaseVision Workshop 2.6)
  - IRIX 6.2 Applications
  - IRIS Power C 6.2

- ORACLE Server 7.3.2
- ORACLE Development 2000 1.3.1
- Network Time Protocol, Version 2.0 (SGI freeware)

The version numbers may change with each software turnover and will be noted in the turnover package.

2.1.1.2 Redundant Array of Inexpensive Devices

2.1.1.2.1 Input RAID

The input RAID on each operational string contains data received from the LGS that is ingested by the raw data capture subsystem (RDCS). The amount and frequency of this data are discussed in Reference 1.

2.1.1.2.2 Output RAID

The output RAID on each operational string contains actual LPS output files. Users may request these files by notifying LPS. Output files are available to users via transfer protocol software.

2.2 Backup String Disks (String 5)

The backup string (string 5) has a system disk, an LPS development (or application) disk, and two RAIDs. The first RAID is reserved for storage of incoming data from LGS. As a development string, the second RAID is reserved for temporary storage of test data sets. As a backup to an operational string, the second RAID is reserved for temporary storage of LPS output data sets awaiting retrieval by the EOS Core System. Figure 2–1 shows the relationship of the backup string to the operational strings (strings 1 to 4).

2.2.1 Logical-to-Physical Disk Mapping of Backup String

This section describes the logical disk allocation defined for the physical disks. Table 2–2 summarizes the allocation of the logical file systems to physical disks.

Table 2–2. Summary of Disk Allocation for Backup String

Disk	File System	Use	Software Allocation
System	/usr	Operations	Same as operational strings
LPS Application	/usr	Development	LPS software
RAID 1	/u01	Front end	Development: Test data only As operational backup: Raw wideband data
RAID 2	/u02	Back end	Development: Test data sets only As operational backup: Level zero R output files

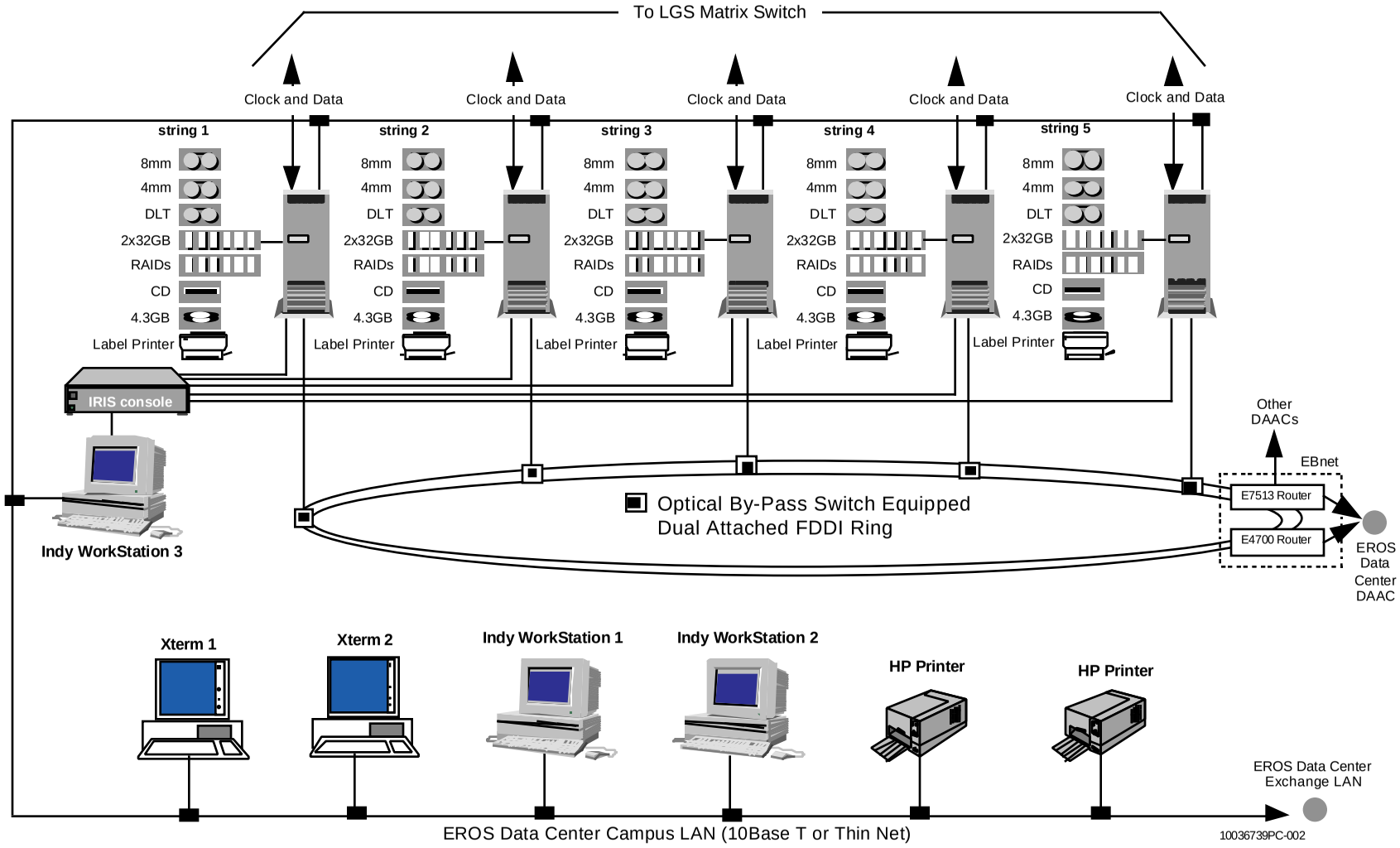


Figure 2–1. Operational Hardware Configuration

2.2.1.1 System Disk (String 5)

At any one time, the system disk on string 5 will contain the following:

- Operating system and associated files
- Three distinct instances of the LPS database, one for each environment (development, test, and operational)
- PVCS library
- Development environment directories (see Section 1.3)
- Government off-the-shelf (GOTS): frame_sync libraries
- COTS
  - Hierarchical data format (HDF) libraries
  - IRIX 6.2 (operating system)
  - NC3/NFS 6.2
  - IRIS Development Option 6.2 (CaseVision Workshop 2.6)
  - IRIX 6.2 Applications
  - IRIS Power C 6.2
  - ORACLE Server 7.3.2
  - ORACLE Development 2000 1.3.1
  - Network Time Protocol, Version 2.0 (SGI freeware)

2.2.1.2 Redundant Array of Inexpensive Devices (String 5)

2.2.1.2.1 Input RAID

At any one time, the input RAID contains raw data files used for testing

- Image Assessment System (IAS) parameters from the calibration parameter file
- Executables that have been modified to provide enhancements to LPS software or to correct errors

2.2.1.2.2 Output RAID

At any one time, the output RAID contains

- Test environment output data files resulting from IAS parameter, other L0R parameters or threshold tests
- Development environment output data files resulting from tests of enhanced or modified software

Section 3. Configuration Management

This section contains the CM directory structure and build process for LPS in the development string. All software written to support the operational LPS will be maintained and controlled by the configuration manager using the CM tool, PVCS Version Manager, which resides on the development string only. All builds must occur in a CM work directory that is outside of PVCS.

3.1 CM Directory Structure in the Development String

All LPS software builds must occur in the CM work directory called *lpswork*. All source and control files under configuration control are stored in the /pvcs/LPS/archives directory. Controlled files include script files and data files needed by certain subsystems. The CM directory structure is illustrated in Figure 3–1. See Appendix B for a sample of the .cshrc file for CM.

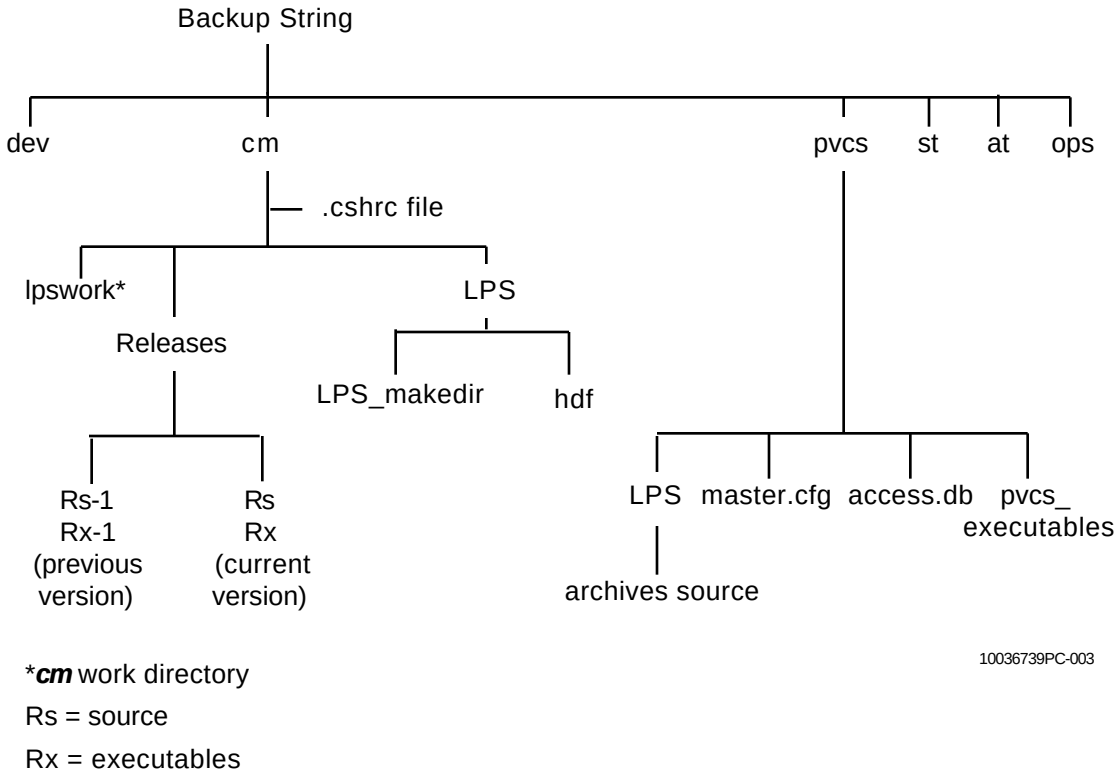


Figure 3–1. CM Directory Structure in the Development String

3.2 Building the System in the Development String

The software written to support LPS is developed, turned over to CM, stored in PVCS, extracted into the CM work directory, and then built for system testing. The executables are copied to operational directories only on approval of system test and operation management.

The process for building a version of LPS is always performed on the development string.

Scripts to build the LPS structure and manage PVCS activities exist and are listed in Section 3.2.1.3.

3.2.1 Preparation for a Build

Preparation for a build involves

- Backup pvcs and current cm directories
- Reviewing the turnover package
- Checking that all COTS and GOTS software are the correct versions and are installed, including the required instances of the database in the test environment
- Making necessary script changes to reflect LPS structural change in the CM work directory and PVCS
- Checking the .cshrc and the .lpsdevrc files for correct environment setups; the .lpsrc is installed with the executables at the test site
- Checking new and modified units into PVCS (Note: Keep a copy of the delivered software because the files are physically moved into PVCS.)

3.2.1.1 Turnover Package

The turnover package contains all build information (a list of new, modified, and deleted units), environment changes, and build instructions.

3.2.1.2 Required COTS and GOTS to Build

Section 2 identifies the COTS and GOTS software required for a build. The correct versions of the COTS and GOTS software should be noted in each build's turnover package. Some of them include the following:

- IRIX 6.2
- ORACLE 7.3.2
- HDF4.Or2
- HDF-EOS1.5v1.00
- Frame_Sync libraries

The usr/cm/LPS subdirectory should contain an hdf directory with appropriate versions of the HDF libraries so that a link can be made to it from the COTS subdirectory in the *usr/cm/lpswork/COTS* directory.

3.2.1.3 Script Changes

All of the following scripts (except LPS_makedir) are located in $HOME/cm/scripts. LPS_makedir is the only PVCS-controlled script at this time (/usr/pvcs/LPS/archives/ LPS_makedir,v), which is extracted from PVCS to the $HOME/cm/LPS directory. Appendix C contains a hardcopy of these scripts. They may require changes if the LPS structure changes.

- new_check: checks units into PVCS
- new_get checks units out of PVCS into the work directory
- LPS_makedir recreates the LPS working directory structure called "*lpswork*" where the build will take place. It also extracts the most current lmake and install scripts and places them in the appropriate subdirectories. The work directory is shown in Figure 3–2. (Table 3–1 provides an explanation of the terms used in Figure 3–2.)
- setup_st_deliv creates the necessary subdirectory structure that will be populated to the test site. It should be updated as required; i.e., build number and other changes.

These scripts change when the LPS structure changes. All scripts should reflect the desired version label.

A copy of these scripts may need to be modified for patches or partial builds.

Any structural modifications to PVCS are made manually.

3.2.1.4 Changes to the LPS_makedir Script

The LPS_makedir script builds the complete LPS tree structure in the CM work directory, *lpswork*; a group label may also be used. It is a controlled script and requires a version label change whenever a new build takes place. LPS structural changes should also be made when necessary. Test the script before returning it to PVCS; this will require either checking in one copy of the modified lmake and install scripts into /usr/pvcs/LPS/archives with the correct version label or adding the correct version label to the current unchanged scripts stored in PVCS.

1. Extract the LPS_makedir script from PVCS:

   cd /usr/cm/LPS

   get -l $PVCS_LPS/archives/LPS_makedir,v

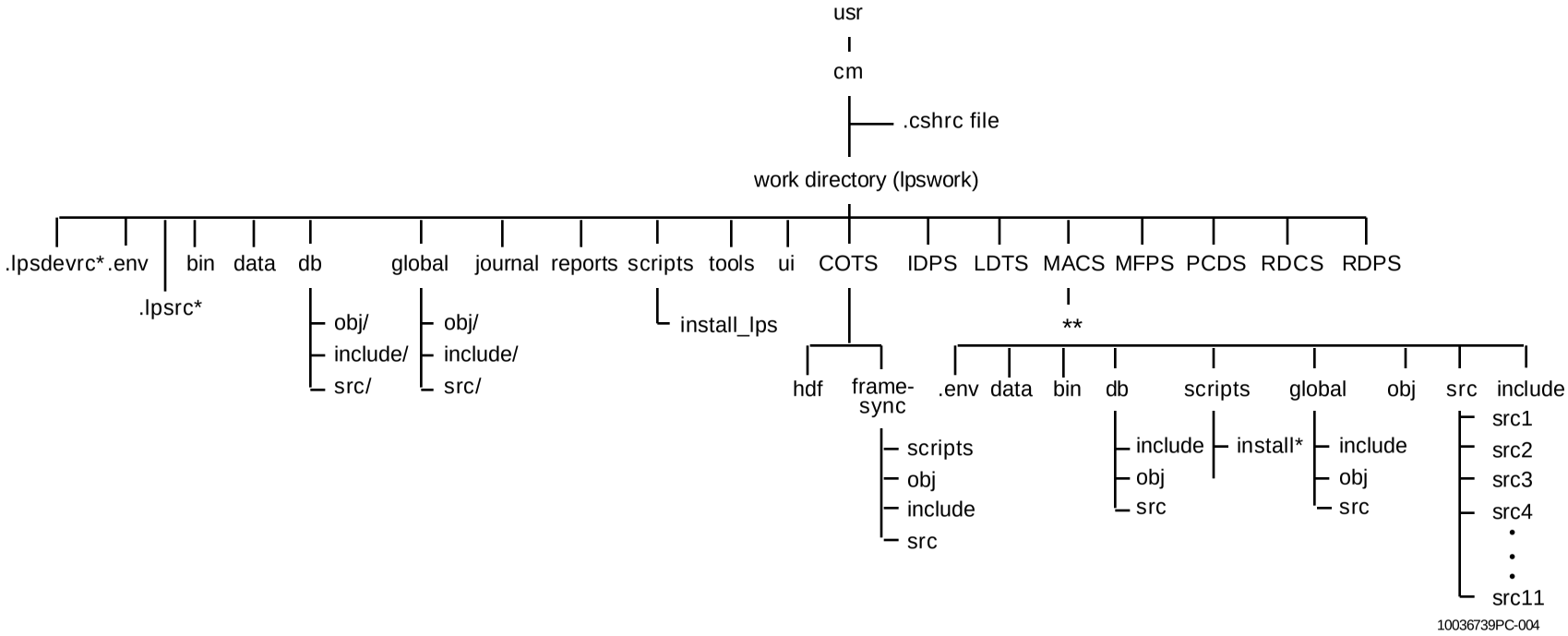
2. Change the version label (i.e., B3.1. or R1.0.) or group label (i.e., ST) for the lmake and install files desired and make the necessary structural changes; B = build and R = release.

3. Save the changed file.

   If the lmake and install scripts are changed, check them into/usr/pvcs/LPS/archives.

   put -vB# or R# /usr/cm/lpswork/install /usr/pvcs/LPS/archives/install,v

   If the lmake and install scripts are unchanged, add the appropriate version label to each unit.



**Each subsystem has all or some of this structure.

Figure 3–2. Development Directory Structure

Table 3–1. Explanation of Terms in Figure 3–2

Term	Explanation
.env	Environment file that holds environment variables for the system and/or subsystem
bin	Holds executables
data	Holds data files; these directories may be subdivided further, perhaps for different contact periods
db	Holds global database routines and include files
global	Holds global routines and global include files; the global directory under the subsystem_name directory contains units common to the different subsystem processes, but not to the project as a whole
include	Holds include files
obj	Holds object files and db and global libraries
scripts	Holds make files and any other scripts
src	Holds source files
src# (# = 1–11)	Holds source files that create an executable; one src# for an executable
subsystem_name	Provides the top-level directory for each subsystem: IDPS, LTDS, MACS, MFPS, PCDS, RDCS, and RDPS
tools	Holds useful tools for developers and maintainers, such as alignments of code program and cadre tools
ui	Holds user interface files
tmp	Originally made available to hold temporary files (may be necessary for some systems); all files in this directory are strictly temporary and are deleted after every system reboot

Back up current lpswork directory if it is to be preserved or remove it:

mv /usr/cm/lpswork bak.lpswork

Or execute the LPS_makedir script to ensure directory structure reflects the correct LPS structure:

cd /usr/cm/LPS

Type **LPS_makedir**

– Return the LPS_makedir to PVCS with the appropriate version label: **put -v[B# or R#] -m"/usr/cm/LPS/LPS_makedir /LPS/*archives*/LPS_makedir,v**

3.2.1.5 Verify User Environment Setup

The CM administrator must have the following statements in the CM .cshrc file to build in the development environment:

· setenv LPS_HOME $HOME/lpswork

- setenv HOME /usr/cm
- setenv PVCSHOME /usr/pvcs
- setenv PATH $PVCSHOME/.:$PATH
- setenv PVCS_LPS /usr/pvcs/LPS
- setenv INIT_HOME $HOME
- setenv CM_HOME $HOME/LPS
- source .lpsdevrc (developers build environment variables)*
- source .lpsrc [test (system test/acceptance test) build environment variables]*

Note: Only csh is supported.

Reference 4 documents the development environment variables for the LPS environment and for each subsystem's environment. Appendix D in Reference 1 describes the variables for the system test and the LPS operating environment.

3.2.1.6 Check Modified and New Units into PVCS

Within the CM directory, all new and modified units are copied in the appropriate subdirectory of the work directory so they can be checked into PVCS. This includes any scripts involved, plus the LPS_makedir script, which will require at least a change in the version label for every build. Only one copy each of the lmake and install scripts is needed in PVCS and should be stored in the /usr/pvcs/LPS/archives directory. The other copies can be removed from the LPS structure. Also .a files are not checked into PVCS.

3.2.2 Detailed Build Process Using the Command Line Version

Once all preparations are complete as described in Section 3.2.1, follow the instructions in the turnover package to perform the following build activities:

- Extract the LPS_makedir script to build the LPS structure tree.
- Extract units from PVCS into the CM work directory for the appropriate release
- Perform the build using the extracted units in the CM work directory.
- Populate the test site(s).

The "new_get" script is executed to extract the appropriate units from PVCS and place them in the appropriate subdirectories in the CM work directory. When the LPS software is appropriately structured and the variables' paths are verified, the build script is executed.

Output reports are generated during a build and are stored in /usr/cm/reports for verification.

3.2.2.1 Build the LPS Directory Structure

The following steps outline how to extract units from PVCS:

1. Login into the /usr/cm account.
2. Extract the LPS_makedir file from PVCS if not already extracted:

   cd /usr/cm/LPS

   get vR2.0 -n $PVCS_LPS/archives/LPS_makedir,v
3. Check that the file is an executable and change the mode it if needed:

   ls -la (to see if file is an executable)

   chmod 775 LPS_makedir (to make a file an executable)
4. Execute the LPS_makedir script to build the LPS work directory structure:

   cd /usr/cm/LPS

   Type **LPS_makedir**
5. When completed, verify the LPS structure against the script.

3.2.2.2 Extract Units From PVCS

1. Extract the appropriate units from PVCS using the "new_get" script.

   NOTE: The new_get script used should have been edited to reflect the correct R#/B# to extract the appropriate units.

   Type **new_get**

   Verify that the correct units have been extracted by comparing the directories against the directory delivered by the developers.

   dircmp - w120 lpswork dev_deliv >& filename
2. Validate the HDF link between /usr/cm/LPS/hdf and/usr/cm/lpswork/COTS/hdf; it is in the LPS_makedir script if needed.

   ls -la /usr/cm/lpswork/COTS

   You should see this line in the list: /usr/cm/lpswork/COTS/hdf --> /usr/cm/LPS/hdf
3. Validate the other environment variables for $HOME, $LPS_HOME, $INIT_HOME.

   echo $INIT_HOME (should be $HOME)

3.2.2.3 Build Executables

Compile LPS as follows:

1. Go to scripts directory in the lpswork structure: cd $LPS_HOME_SCRIPTS
2. Create a script file to record the build; type **script filename.tscr**
3. Execute the build script: **install_lps nodebug**
4. Once the build is complete, type "exit" to discontinue recording the build
5. Check the
   - Executables in /usr/cm/lpswork/bin against the list of executables in the turnover/delivery package
   - File (filename.tscr) for error messages: **MORE filename.tscr lgrep "ERROR:"** (other keywords to check are "unable," "cannot," "can't", and "don't know")
   - Build instructions for any special setups required for system test, such as data files or files required to be in the system test's $HOME account

3.2.2.4 Populate to Test Site

1. Prepare the structure that will be populated to the test site by using the setup_st_deliv script. It creates the required directories and copies directories containing executables and files (including Idp_mwd and Tk2Motif) to a b#_4_st directory. Appendix C provides a hardcopy of the script.
   - Edit the script. Change the build number "b#" in the line "mkdir b#_4_st" and add or delete any new subdirectories or special files as needed.
   - In the $HOME directory, execute the setup_st_deliv script. Figure 3–3 shows the directory structure of a populated test site environment.
2. Copy the b#_4_st directory to the test site (i.e., /usr/LPS/st) and verify the sizes of the files copied:

   cp -pr b#_4_st <test directory path> (-p preserves the date)
3. Edit the .cshrc file and set the LPS_HOME path to the appropriate build (b) or release (r) number:

   setenv LPS_HOME /usr/LPS/st/b#
4. Rename b#_4_st to the appropriate build (b) or release (r) number:

   cd /usr/LPS/st

   mv b#_4_st b#
5. Edit the .lpsrc file:

   Find and change any reference to b# to the appropriate build number (b#). Search for the line "setenv LPS_HOME" and edit the build or release number.

Change the ORACLE_HOME path in the .lpsrc file:

setenv ORACLE_HOME /usr/app/oracle/client/developer2000/1.3.1

6. Make sure that testers move the Idp_mwd and Tk2Motif files to their HOME directory.

7. Verify that the environment variables and paths are correct before testing begins. Appendix D provides a sample of the environment variables for system test.

   cd b# and source .lpsrc

   echo $LPS_HOME (should be usr/LPS/st/b#)

   echo $LPS_RAWFILE_PATH (should be u01/at/r2/rawfile)

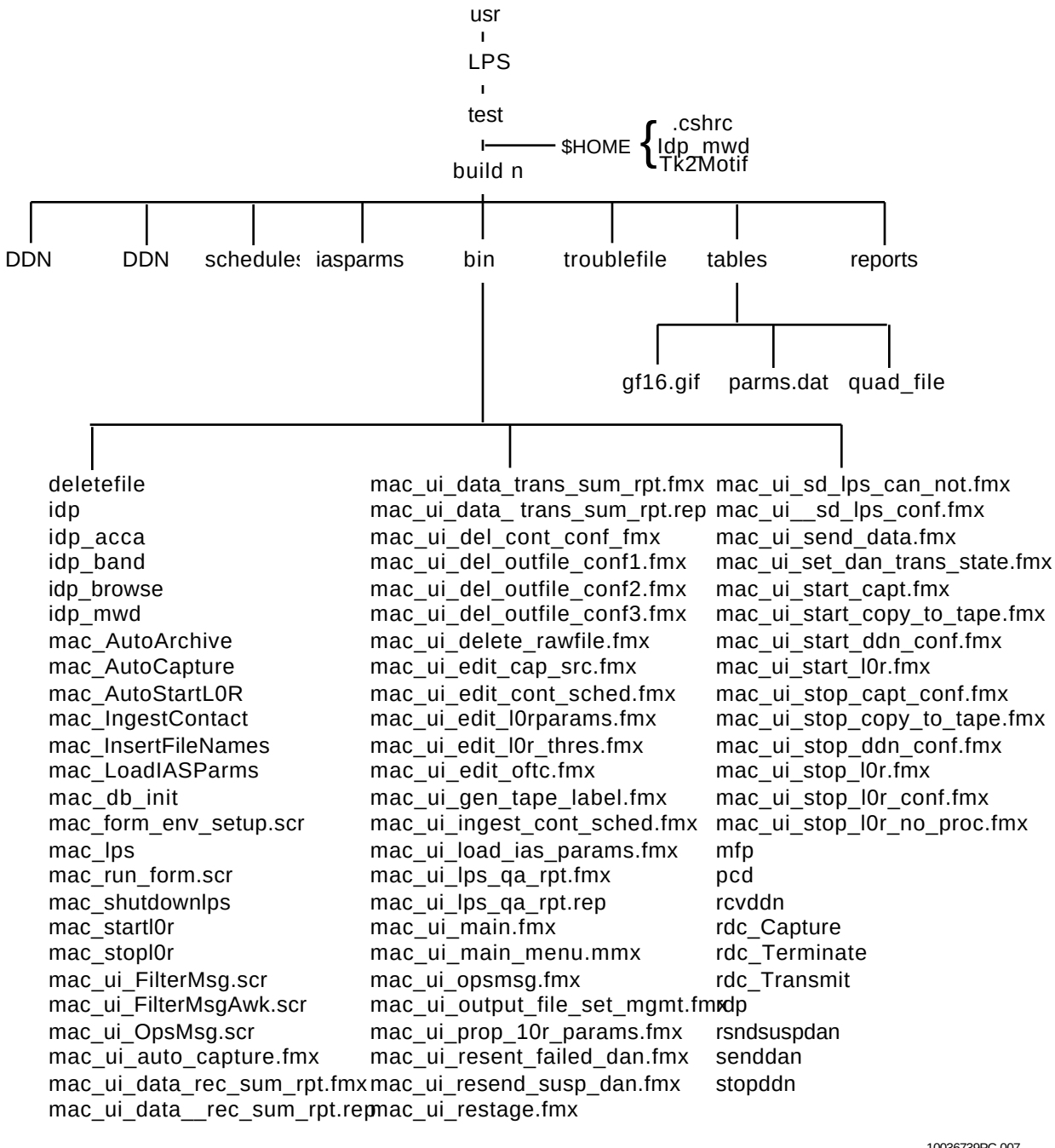


Figure 3–3. Development Test Directory Structure

echo $LPS_OUTFILE_PATH (should be u02/at/r2/outfile)

echo $ORACLE_HOME (/usr/app/oracle/client/developer2000/1.3.1)

echo $LPS_TAPE_DEV (/dev/rmt/tps131d5)

echo $LPS_TAPE_LIBRARY_DEV (/dev/scsi/sc131d5l0)

8. Once all verification is complete, change file ownership:

   cd /usr/LPS/st

   chmod 775 b#

   chown -R lpsst.st st

9. Have the system administrator change ownership of three RDCS executables and set user ID as follows:

   chown root:sys rdc_Capture rdc_Transmit rdc_DeleteFiles

   chmod u+s rdc_Capture rdc_Transmit rdc_DeleteFiles

Testing can begin. See Appendix E for a sample of the .cshrc file for lpsst.

3.2.3 Building for Operations

Because all source code is kept in PVCS on the backup string (string 5), the software builds for the operational environment are performed in the CM work directory on the backup string. When the build is complete and accepted by the testers and operations management, the executables are transferred to the operational site. The operational directory structure is shown in Figure 3–4.

3.3 Populating the Runtime Directories

This section discusses the process of populating the runtime directories for the development system and the operational system once a build has been completed.

3.3.1 Development System

A runtime environment directory structure (Figure 3–5) exists on all LPS development hardware elements. This structure is identical for each element and contains identical directory structures for development, CM, and system test/acceptance test (Figure 3–3). The process for populating the runtime directories is the same for each test group as described in Section 3.2.2.4.

3.3.2 Operational System

A build script, setup_ops, is available to create runtime directory structures on all operational hardware elements for the current release if those structures do not exist. The executables are copied or transferred via FTP to the /usr/LPS/ops directory on each operational string, where they will be available to each operational hardware element. The structure is the shown in Figure 3–6.

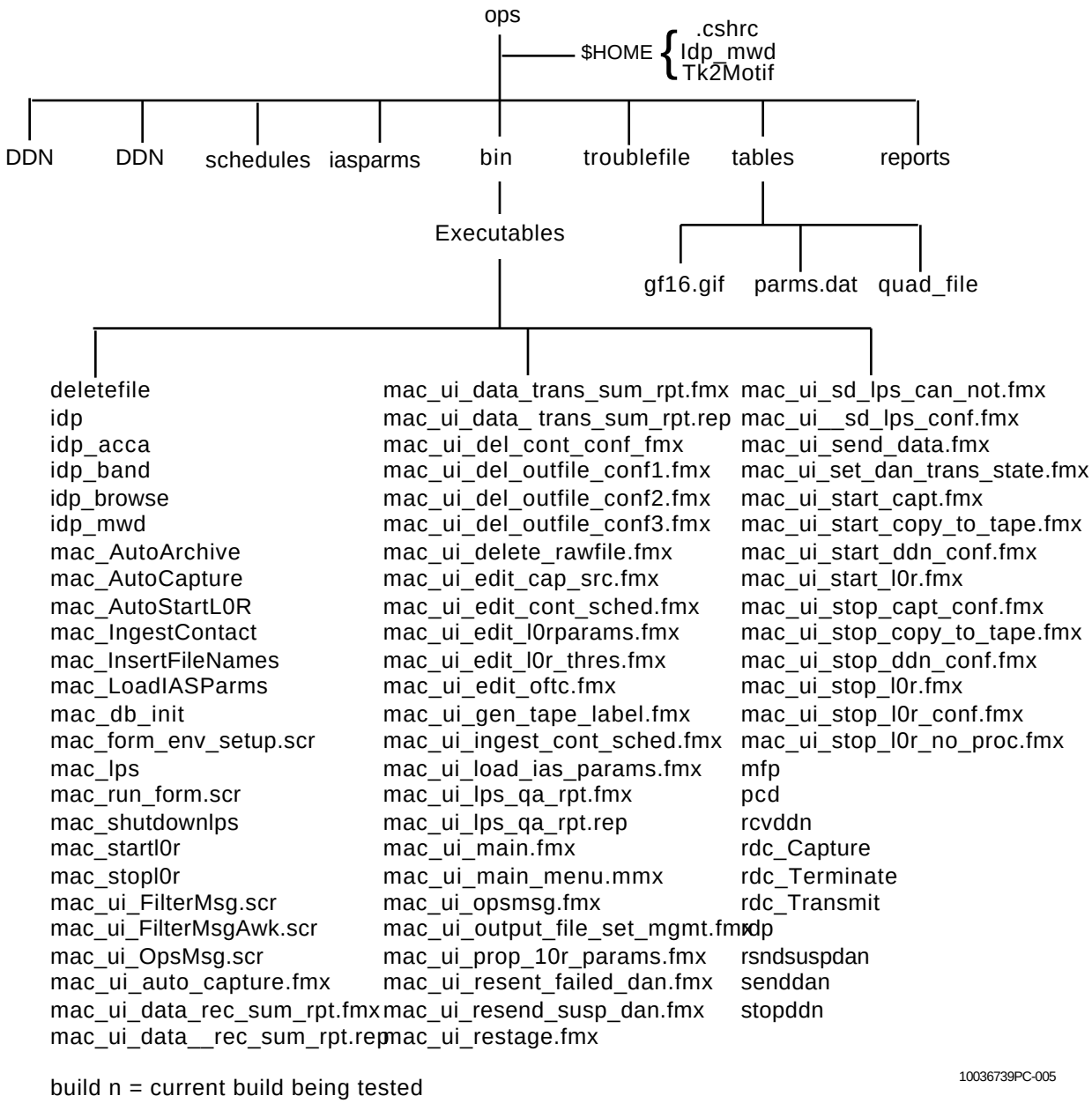
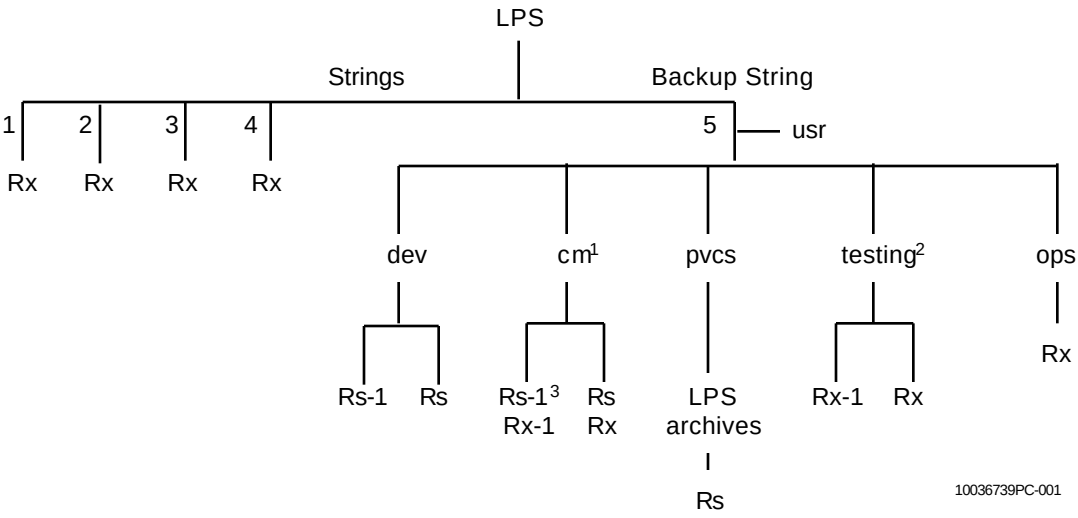


Figure 3–4. Operations Directory Structure



Rs = current release's source for that environment

Rs-1 = previous release's source for that environment

Rx = current release's executables for that environment

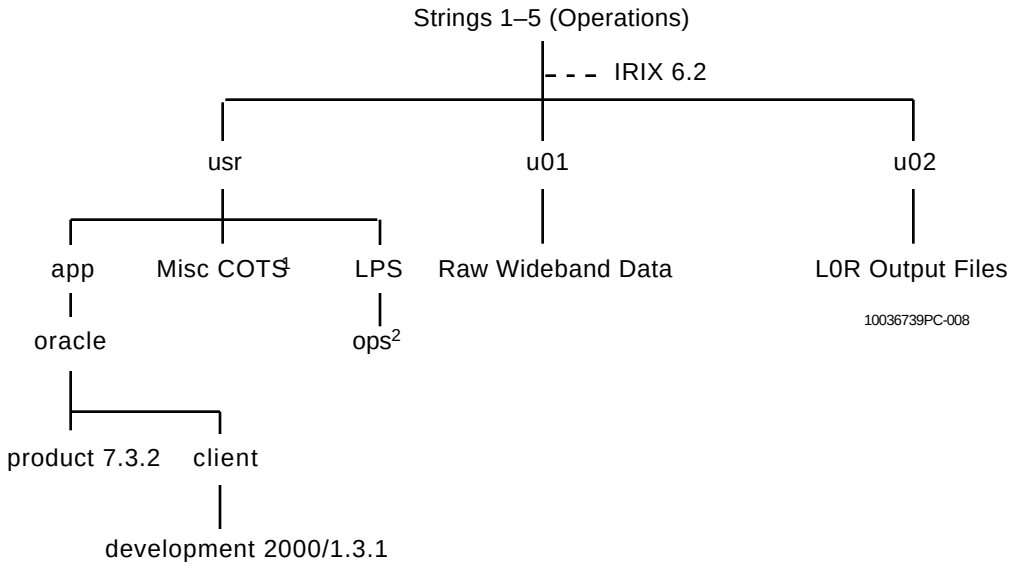
Rx-1 = previous release's executables for that environment

[1]See Figure 3–1

[2]See Figure 3–2

[3]See Figure 3–3

Figure 3–5. Development Runtime Directory Structure



[1]Installation of the operating system, IRIX 6.2, automatically populates the miscellaneous COT

[2]Current version of the executables; see Figure 3–4 for the operational string structure.

Figure 3–6. Operational Runtime Directory Structure

Once the system is accepted by test and operations, the executables are populated to the operational strings. The executables are to be populated via FTP to strings 1–5 and copied to string 5, /usr/LPS/ops.

- To populate to strings 1–4:

  Prepare to send the build/release via FTP to another machine:

  in lpsdev1 (string 5),

  cd /usr/LPS/st/b#

  TAR: tar cvf <b#r#ops.tar filename > < b#r#>& outfilename

  Compress: Compress <tar file> (becomes .Z file)

- In another screen, login to the target operational machine:

  telnet lps00#

  cd /usr/LPS/ops

  ftp lpsdev1

  cd /usr/LPS/st

  get r#tarfile.Z

  bye

- Uncompress the .Z file.
- Untar the .tar file: tar xvf b#r#.tar >$ b#r#tar.log.
- Verify the executables and environment variables.

3.4 LPS Release Types

There are four types of LPS releases:

1. **Release** – Involves major functionality changes, a complete system rebuild, and goes through a complete testing cycle. Planned well in advance. (See Section 3.2.)
2. **Mini-Release** – Considered a maintenance release, and involves minor functionality changes. It can require either a full or partial rebuild, and goes through a complete testing cycle. Planned as a follow-on to a release.
3. **Patch** – Considered a means to provide critical functionality or correct mission-critical problems when sufficient time exists for a complete testing cycle. Involves a partial system rebuild.

4. **Engineering Mode** – Considered a means to provide mission-critical functionality when there is no time for a complete testing cycle. Only limited module, integration, and system testing is performed. Executables only are provided to expedite resolution of the problem. Source code and fill testing of the functionality is provided in the next scheduled release, mini-release, or patch of the system.

3.4.1 Build Notes from a Mini_Release or Patch

CM retains a current copy of the LPS directory structure and source outside of PVCS to save time if a patch or mini release is required. It is copied to lpswork.

The delivered or changed units are checked into PVCS.

In lpswork, delete the old unit(s). Check out the changed unit(s) from PVCS into the appropriate directory(s).

Follow the build instruction that comes with the delivered units.

3.4.2 Build Notes for an Engineering Mode

No units are to be checked into PVCS when an engineering build occurs. Follow the build instructions to perform this type of build.

Appendix A. Checklist for LPS Software Installation

The system administrator should perform the backup restoration or installation up to step 4; CM performs steps 5 and 6; and system test performs steps 7 and 8.

☐ Use backup tape to restore system.

OR

☐ 1. Install IRIX 6.2 Operating System (see Appendix F).*

☐ 2. Install Network Time Protocol (see Appendix G).

☐ 3. Set up directory structure (see Figure 1–1).

☐ 4. Install COTS (see Section 2).*

☐ Install ORACLE Server and database instance (see Appendix H).

☐ Install HDF on string 5 only (follow vendor installation procedures to install the required version or use tape provided).

☐ 5. Install LPS executables.

If not backed up, you will have to extract the R2.0 units from PVCS and compile to create the executables (see Section 3).

☐ 6. Verify environment variables and paths (see Appendix I).

☐ 7. Perform regression tests.

☐ 8. Run LPS.

*Refer to vendor installation procedures.

Appendix B. Sample Configuration Management's .cshrc File

```
###############################################################################
# Sample .cshrc for LPS users/developers                                      #
###############################################################################
set prompt=lpsdev1:"%/ ->"

# Set up for remote protocol with other servers
if ($?USER == 0 || $?prompt == 0) exit

set path = (. ~/bin   /usr/bin  /usr/bsd \
              /etc  /usr/include  /usr/include/sys \
             /usr/local /usr/local/bin  /usr/sbin )

setenv LPS_HOME $HOME/lpswork
setenv CM_HOME $HOME/LPS
setenv INIT_HOME $HOME
setenv DELIV /u04/cm/Dev_Deliveries/r2pvcs/r2.0
setenv LOGS  /u04/cm/LOGS
setenv DEV /u04/cm/Dev_Deliveries/

# Add LPS environment settings
if (-e $LPS_HOME ) then
  # Force a clean path (recommended)
  unsetenv PATH
  unsetenv MANPATH
  unsetenv LD_LIBRARY_PATH

  # Set the default path and sets up LPS operations/test environment
  # Developers/maintenance programmers use .lpsdevrc instead of .lpsrc
  source $LPS_HOME/.lpsdevrc
# source $LPS_HOME/.lpsrc
else
  echo "LPS_HOME is not available."
  setenv PATH /usr/local/bin:/usr/bin:/bin:/usr/etc:/etc:/usr/bsd:/usr/sbin:/sbin:/usr/include/:
endif

#Setup for PVCS
setenv PVCSHOME /usr/pvcs
setenv PATH $PVCSHOME/.:$PATH
setenv PVCS_LPS /usr/pvcs/LPS

# Add my own environment
setenv PATH .:$HOME/bin:$PATH

# File completion using the escape but ignoring certain extensions.
set filec
set fignore=.o
set noclobber
set ignoreeof

# Set the history
set history=40

# Set the prompt
set prompt="`uname -n`:`whoami`:%/[%!\] %"

#       settings for interactive shells

if ($shell == /bin/csh) then
```

```
  ##### PROMPT FOR csh
  set prompt="`uname -n`:`whoami`:%/[%!\] %"

  ##### cd ALIAS FOR csh
  alias cd 'cd \!*;set prompt = "`whoami`@`uname -n`(`uname`):$cwd"\[\!\]"=> "date'
endif

if ($shell == /bin/tcsh) then
  ##### PROMPT FOR tcsh
  set prompt="`hostname`:%/[%!\] %"

  ##### ENABLE vi COMMAND LINE EDITING FOR tcsh
  #####    To use emacs command line editing substitute "emacs" for
  #####    "vi" on the next line
  bind vi
endif

# Set the backspace key (Ctrl-v then backspace)
if ($TERM == sun-cmd || $TERM == xterm) then
  stty echo erase
endif

# Initialize all aliases
if (-e $HOME/.alias) then
  source $HOME/.alias
endif

set history=1000
set notify
set savehist=300
```

Appendix C. Configuration Management Scripts

This appendix provides copies of the following CM scripts:

1. new_check – checks units into PVCS
2. new_get – extracts units from PVCS
3. LPS_makedir – extracts lmake and install from pvcs, creates the LPS directory structure, copies the lmake and install files in directories, and sets up the hdf link between the $HOME/LPS/hdf and $LPS_HOME/COTS.
4. setup_st_deliv – creates the directory structure to be installed for system test.

C.1 new_check

```
#**************************************
#put script – to put all of LPS into PVCS once the archive (header)is created.
#
#Changes:
#1/24/97 cm: – added /bin dir contained .scr files; added to this script.
#          – LDTS include and global lines fixed
#               – added MFPS/data subdir for parms.dat
#-1/29: added MFPS/scripts
#-Add PCDS dirs
#-4/23/97 – edited in additions
#-6/2/97 – added bin to store rdc_Terminate
#-7/2/97 – added for B3.2 IDPS/src2..src4, LDTS/src4, MACS/src5..src10
#-7/5/97 – Commented out RDCS/src1 LPS_makedir does not create it until
#-7/6/9 – B3.2 dirs added: RDCS/src/src1..src9, MACS/src/src11
#************************************************************

#!/bin/sh

echo “Working in the bin directory”
cd /u04/cm/lpswork/bin
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/bin/*,v

echo “Working in the COTS directory”
cd /u04/cm/lpswork/COTS
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/COTS/archives/.*,v
cd /u04/cm/lpswork/COTS/frame_sync/src
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/COTS/frame_sync/src/*,v
cd /u04/cm/lpswork/COTS/frame_sync/include
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/COTS/frame_sync/include/*,v

echo “Working in the IDPS directory”
cd /u04/cm/lpswork/IDPS
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/IDPS/archives/.*,v
cd /u04/cm/lpswork/IDPS/data
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/IDPS/data/.*,v
cd /u04/cm/lpswork/IDPS/global/include
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/IDPS/global/include/*,v
cd /u04/cm/lpswork/IDPS/global/src
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/IDPS/global/src/*,v
cd /u04/cm/lpswork/IDPS/db/include
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/IDPS/db/include/*,v
cd /u04/cm/lpswork/IDPS/db/src
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/IDPS/db/src/*,v
```

```
cd /u04/cm/lpswork/IDPS/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/IDPS/include/*,v
cd /u04/cm/lpswork/IDPS/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/IDPS/src/*,v
cd /u04/cm/lpswork/IDPS/src/src1
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/IDPS/src/src1/*,v
cd /u04/cm/lpswork/IDPS/src/src2
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/IDPS/src/src2/*,v
cd /u04/cm/lpswork/IDPS/src/src3
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/IDPS/src/src3/*,v
cd /u04/cm/lpswork/IDPS/src/src4
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/IDPS/src/src4/*,v
cd /u04/cm/lpswork/IDPS/scripts
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/IDPS/scripts/*,v

echo "Working in the LDTS directory"
cd /u04/cm/lpswork/LDTS
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/archives/.*,v
cd /u04/cm/lpswork/LDTS/data
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/data/*,v
cd /u04/cm/lpswork/LDTS/db/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/db/src/*,v
cd /u04/cm/lpswork/LDTS/db/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/db/include/*,v
cd /u04/cm/lpswork/LDTS/global/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/global/src/*,v
cd /u04/cm/lpswork/LDTS/global/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/global/include/*,v
cd /u04/cm/lpswork/LDTS/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/include/*,v
cd /u04/cm/lpswork/LDTS/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/src/*,v
cd /u04/cm/lpswork/LDTS/src/src1
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/src/src1/*,v
cd /u04/cm/lpswork/LDTS/src/src2
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/src/src2/*,v
cd /u04/cm/lpswork/LDTS/src/src3
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/src/src3/*,v
cd /u04/cm/lpswork/LDTS/src/src4
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/src/src4/*,v
cd /u04/cm/lpswork/LDTS/scripts
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/LDTS/scripts/*,v

echo "Working in the MACS directory"
cd /u04/cm/lpswork/MACS
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/archives/.*,v
cd /u04/cm/lpswork/MACS/data
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/data/*,v
cd /u04/cm/lpswork/MACS/db/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/db/include/*,v
cd /u04/cm/lpswork/MACS/db/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/db/src/*,v
cd /u04/cm/lpswork/MACS/global/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/global/include/*,v
cd /u04/cm/lpswork/MACS/global/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/global/src/*,v
cd /u04/cm/lpswork/MACS/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/include/*,v
cd /u04/cm/lpswork/MACS/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/*,v
cd /u04/cm/lpswork/MACS/src/src1
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/src1/*,v
cd /u04/cm/lpswork/MACS/src/src2
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/src2/*,v
```

```
cd /u04/cm/lpswork/MACS/src/src3
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/src3/*,v
cd /u04/cm/lpswork/MACS/src/src4
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/src4/*,v
cd /u04/cm/lpswork/MACS/src/src5
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/src5/*,v
cd /u04/cm/lpswork/MACS/src/src6
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/src6/*,v
cd /u04/cm/lpswork/MACS/src/src7
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/src7/*,v
cd /u04/cm/lpswork/MACS/src/src8
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/src8/*,v
cd /u04/cm/lpswork/MACS/src/src9
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/src9/*,v
cd /u04/cm/lpswork/MACS/src/src10
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/src10/*,v
cd /u04/cm/lpswork/MACS/src/src11
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/src11/*,v
cd /u04/cm/lpswork/MACS/src/src11put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/src/src10/*,v

cd /u04/cm/lpswork/MACS/scripts
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MACS/scripts/*,v

echo "Working in the MFPS directory"
cd /u04/cm/lpswork/MFPS
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MFPS/archives/.*,v
cd /u04/cm/lpswork/MFPS/data
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MFPS/data/*,v
cd /u04/cm/lpswork/MFPS/db/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MFPS/db/src/*,v
cd /u04/cm/lpswork/MFPS/global/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MFPS/global/include/*,v
cd /u04/cm/lpswork/MFPS/global/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MFPS/global/src/*,v
cd /u04/cm/lpswork/MFPS/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MFPS/include/*,v
cd /u04/cm/lpswork/MFPS/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MFPS/src/*,v
cd /u04/cm/lpswork/MFPS/scripts
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/MFPS/scripts/*,v

echo "Working in the PCDS directory"
cd /u04/cm/lpswork/PCDS
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/PCDS/archives/.*,v
cd /u04/cm/lpswork/PCDS/data
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/PCDS/data/*,v
cd /u04/cm/lpswork/PCDS/db
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/PCDS/db/*,v
cd /u04/cm/lpswork/PCDS/db/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/PCDS/db/include/*,v
cd /u04/cm/lpswork/PCDS/db/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/PCDS/db/src/*,v
cd /u04/cm/lpswork/PCDS/global/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/PCDS/global/include/*,v
cd /u04/cm/lpswork/PCDS/global/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/PCDS/global/src/*,v
cd /u04/cm/lpswork/PCDS/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/PCDS/include/*,v
cd /u04/cm/lpswork/PCDS/scripts
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/PCDS/scripts/*,v
cd /u04/cm/lpswork/PCDS/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/PCDS/src/*,v

echo "Working in the RDCS directory"
```

```
cd /u04/cm/lpswork/RDCS
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/archives/.*,v
cd /u04/cm/lpswork/RDCS/data
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/data/*,v
cd /u04/cm/lpswork/RDCS/bin
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/bin/*,v
cd /u04/cm/lpswork/RDCS/db/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/db/src/*,v
cd /u04/cm/lpswork/RDCS/global/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/global/include/*,v
cd /u04/cm/lpswork/RDCS/global/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/global/src/*,v
cd /u04/cm/lpswork/RDCS/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/include/*,v
cd /u04/cm/lpswork/RDCS/scripts
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/scripts/*,v
cd /u04/cm/lpswork/RDCS/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/src/*,v
cd /u04/cm/lpswork/RDCS/src/src1
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/src/src1/*,v
cd /u04/cm/lpswork/RDCS/src/src2
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/src/src2/*,v
cd /u04/cm/lpswork/RDCS/src/src3
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/src/src3/*,v
cd /u04/cm/lpswork/RDCS/src/src4
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/src/src4/*,v
cd /u04/cm/lpswork/RDCS/src/src5
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/src/src5/*,v
cd /u04/cm/lpswork/RDCS/src/src6
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/src/src6/*,v
cd /u04/cm/lpswork/RDCS/src/src7
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/src/src7/*,v
cd /u04/cm/lpswork/RDCS/src/src8
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/src/src8/*,v
cd /u04/cm/lpswork/RDCS/src/src9
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDCS/src/src9/*,v

echo "Working in the RDPS directory"
cd /u04/cm/lpswork/RDPS
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDPS/archives/.*,v
cd /u04/cm/lpswork/RDPS/data
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDPS/data/*,v
cd /u04/cm/lpswork/RDPS/db/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDPS/db/src/*,v
cd /u04/cm/lpswork/RDPS/global/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDPS/global/include/*,v
cd /u04/cm/lpswork/RDPS/global/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDPS/global/src/*,v
cd /u04/cm/lpswork/RDPS/include
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDPS/include/*,v
cd /u04/cm/lpswork/RDPS/src
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDPS/src/*,v
cd /u04/cm/lpswork/RDPS/scripts
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/RDPS/scripts/*,v

echo "Working in the lpswork directory"
cd /u04/cm/lpswork
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/archives/.*,v
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/archives/*,v

#scripts that extract from RCS (e.g. tco)
cd /u04/cm/lpswork/RCS_scripts
put -vB3.2 -n -m"B3.2 Delivered Units" /u04/pvcs/LPS/archives/*,v
```

```
echo “Working in the data directory”
cd /u04/cm/lpswork/data
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/data/*,v

echo “Working in the db directory”
cd /u04/cm/lpswork/db/include
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/db/include/*,v
cd /u04/cm/lpswork/db/src
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/db/src/*,v

echo “Working in the global directory”
cd /u04/cm/lpswork/global/include
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/global/include/*,v
cd /u04/cm/lpswork/global/src
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/global/src/*,v

echo “Working in the scripts directory”
cd /u04/cm/lpswork/scripts
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/scripts/*,v

echo “Working in the tables directory”
cd /u04/cm/lpswork/tables
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/tables/*,v

echo “Working in the ui directory”
cd /u04/cm/lpswork/ui/src
put -vB3.2 -n -m”B3.2 Delivered Units” /u04/pvcs/LPS/ui/src/*,v
```

C.2 new_get

```
#************************************
#new_check script: to get units from PVCS for b31
#  If the directory structure changes, copy the new_check to new_get
# so structure is the same else just change the v#.

#
#
#1/24/97 cm: – added /bin dir contained .scr files; added to this script.
#          – LDTS include and global lines fixed
#               – added MFPS/data subdir for parms.dat
#-1/29: added MFPS/scripts
#-Add PCDS dirs
#-4/23/97 – edited in additions
#-6/2/97 – added bin to store rdc_Terminate
#-7/2/97 – added for B3.1 IDPS/src2..src4, LDTS/src4, MACS/src5..src10
#-7/5/97 – For B3.1:Commented out RDCS/src1 LPS_makedir does not create it until
#                    B3.2 and copied b31new_check to b31new_get
#        Edited the get string a get command:
#         i.e.:  get -vB3.2 -n $PVCS_LPS/MFPS/data/*,v
#              Only need .files from PVCS/LPS/archives (/.*,v) ;so,
#                        commented out the PVCS/LPS/archives/*,v and the
#              RCS_scripts contain the tco and misc RCS files.
# 7/14/97 – Edited in macs(src11) and RDCS(src1-src9) for b3.2
# 7/15/97 – changed B3.1 to B3.2 for test of getb3.2
# 7/17/97 – changed B3.2 to R2.0
# 7/24/97 – added .tco,v  extract*,v  and makedir*,v to the get RCS scripts
#           section to prevent the lmake, install and LPS_makedir
#           from being extracted a 2nd time.  The RCS scripts
#           are no longer delivered after bld3.1.
#************************************************************

#!/bin/sh
```

```
echo “Working in the bin directory”
cd /u04/cm/lpswork/bin
get -vR2.0 -n $PVCS_LPS/bin/*,v

echo “Working in the COTS directory”
cd /u04/cm/lpswork/COTS
get -vR2.0 -n $PVCS_LPS/COTS/archives/.*,v
cd /u04/cm/lpswork/COTS/frame_sync/src
get -vR2.0 -n $PVCS_LPS/COTS/frame_sync/src/*,v
cd /u04/cm/lpswork/COTS/frame_sync/include
get -vR2.0 -n $PVCS_LPS/COTS/frame_sync/include/*,v

echo “Working in the IDPS directory”
cd /u04/cm/lpswork/IDPS
get -vR2.0 -n $PVCS_LPS/IDPS/archives/.*,v
cd /u04/cm/lpswork/IDPS/data
get -vR2.0 -n $PVCS_LPS/IDPS/data/.*,v
cd /u04/cm/lpswork/IDPS/global/include
get -vR2.0 -n $PVCS_LPS/IDPS/global/include/*,v
cd /u04/cm/lpswork/IDPS/global/src
get -vR2.0 -n $PVCS_LPS/IDPS/global/src/*,v
cd /u04/cm/lpswork/IDPS/db/include
get -vR2.0 -n $PVCS_LPS/IDPS/db/include/*,v
cd /u04/cm/lpswork/IDPS/db/src
get -vR2.0 -n $PVCS_LPS/IDPS/db/src/*,v
cd /u04/cm/lpswork/IDPS/include
get -vR2.0 -n $PVCS_LPS/IDPS/include/*,v
cd /u04/cm/lpswork/IDPS/src
get -vR2.0 -n $PVCS_LPS/IDPS/src/*,v
cd /u04/cm/lpswork/IDPS/src/src1
get -vR2.0 -n $PVCS_LPS/IDPS/src/src1/*,v
cd /u04/cm/lpswork/IDPS/src/src2
get -vR2.0 -n $PVCS_LPS/IDPS/src/src2/*,v
cd /u04/cm/lpswork/IDPS/src/src3
get -vR2.0 -n $PVCS_LPS/IDPS/src/src3/*,v
cd /u04/cm/lpswork/IDPS/src/src4
get -vR2.0 -n $PVCS_LPS/IDPS/src/src4/*,v
cd /u04/cm/lpswork/IDPS/scripts
get -vR2.0 -n $PVCS_LPS/IDPS/scripts/*,v

echo “Working in the LDTS directory”
cd /u04/cm/lpswork/LDTS
get -vR2.0 -n $PVCS_LPS/LDTS/archives/.*,v
cd /u04/cm/lpswork/LDTS/data
get -vR2.0 -n $PVCS_LPS/LDTS/data/*,v
cd /u04/cm/lpswork/LDTS/db/src
get -vR2.0 -n $PVCS_LPS/LDTS/db/src/*,v
cd /u04/cm/lpswork/LDTS/db/include
get -vR2.0 -n $PVCS_LPS/LDTS/db/include/*,v
cd /u04/cm/lpswork/LDTS/global/src
get -vR2.0 -n $PVCS_LPS/LDTS/global/src/*,v
cd /u04/cm/lpswork/LDTS/global/include
get -vR2.0 -n $PVCS_LPS/LDTS/global/include/*,v
cd /u04/cm/lpswork/LDTS/include
get -vR2.0 -n $PVCS_LPS/LDTS/include/*,v
cd /u04/cm/lpswork/LDTS/src
get -vR2.0 -n $PVCS_LPS/LDTS/src/*,v
cd /u04/cm/lpswork/LDTS/src/src1
get -vR2.0 -n $PVCS_LPS/LDTS/src/src1/*,v
cd /u04/cm/lpswork/LDTS/src/src2
get -vR2.0 -n $PVCS_LPS/LDTS/src/src2/*,v
cd /u04/cm/lpswork/LDTS/src/src3
get -vR2.0 -n $PVCS_LPS/LDTS/src/src3/*,v
cd /u04/cm/lpswork/LDTS/src/src4
```

```
get -vR2.0 -n $PVCS_LPS/LDTS/src/src4/*,v
cd /u04/cm/lpswork/LDTS/scripts
get -vR2.0 -n $PVCS_LPS/LDTS/scripts/*,v

echo “Working in the MACS directory”
cd /u04/cm/lpswork/MACS
get -vR2.0 -n $PVCS_LPS/MACS/archives/.*,v
cd /u04/cm/lpswork/MACS/data
get -vR2.0 -n $PVCS_LPS/MACS/data/*,v
cd /u04/cm/lpswork/MACS/db/include
get -vR2.0 -n $PVCS_LPS/MACS/db/include/*,v
cd /u04/cm/lpswork/MACS/db/src
get -vR2.0 -n $PVCS_LPS/MACS/db/src/*,v
cd /u04/cm/lpswork/MACS/global/include
get -vR2.0 -n $PVCS_LPS/MACS/global/include/*,v
cd /u04/cm/lpswork/MACS/global/src
get -vR2.0 -n $PVCS_LPS/MACS/global/src/*,v
cd /u04/cm/lpswork/MACS/include
get -vR2.0 -n $PVCS_LPS/MACS/include/*,v
cd /u04/cm/lpswork/MACS/src
get -vR2.0 -n $PVCS_LPS/MACS/src/*,v
cd /u04/cm/lpswork/MACS/src/src1
get -vR2.0 -n $PVCS_LPS/MACS/src/src1/*,v
cd /u04/cm/lpswork/MACS/src/src2
get -vR2.0 -n $PVCS_LPS/MACS/src/src2/*,v
cd /u04/cm/lpswork/MACS/src/src3
get -vR2.0 -n $PVCS_LPS/MACS/src/src3/*,v
cd /u04/cm/lpswork/MACS/src/src4
get -vR2.0 -n $PVCS_LPS/MACS/src/src4/*,v
cd /u04/cm/lpswork/MACS/src/src5
get -vR2.0 -n $PVCS_LPS/MACS/src/src5/*,v
cd /u04/cm/lpswork/MACS/src/src6
get -vR2.0 -n $PVCS_LPS/MACS/src/src6/*,v
cd /u04/cm/lpswork/MACS/src/src7
get -vR2.0 -n $PVCS_LPS/MACS/src/src7/*,v
cd /u04/cm/lpswork/MACS/src/src8
get -vR2.0 -n $PVCS_LPS/MACS/src/src8/*,v
cd /u04/cm/lpswork/MACS/src/src9
get -vR2.0 -n $PVCS_LPS/MACS/src/src9/*,v
cd /u04/cm/lpswork/MACS/src/src10
get -vR2.0 -n $PVCS_LPS/MACS/src/src10/*,v
cd /u04/cm/lpswork/MACS/src/src11
get -vR2.0 -n $PVCS_LPS/MACS/src/src11/*,v
cd /u04/cm/lpswork/MACS/scripts
get -vR2.0 -n $PVCS_LPS/MACS/scripts/*,v

echo “Working in the MFPS directory”
cd /u04/cm/lpswork/MFPS
get -vR2.0 -n $PVCS_LPS/MFPS/archives/.*,v
cd /u04/cm/lpswork/MFPS/data
get -vR2.0 -n $PVCS_LPS/MFPS/data/*,v
cd /u04/cm/lpswork/MFPS/db/src
get -vR2.0 -n $PVCS_LPS/MFPS/db/src/*,v
cd /u04/cm/lpswork/MFPS/global/include
get -vR2.0 -n $PVCS_LPS/MFPS/global/include/*,v
cd /u04/cm/lpswork/MFPS/global/src
get -vR2.0 -n $PVCS_LPS/MFPS/global/src/*,v
cd /u04/cm/lpswork/MFPS/include
get -vR2.0 -n $PVCS_LPS/MFPS/include/*,v
cd /u04/cm/lpswork/MFPS/src
get -vR2.0 -n $PVCS_LPS/MFPS/src/*,v
cd /u04/cm/lpswork/MFPS/scripts
get -vR2.0 -n $PVCS_LPS/MFPS/scripts/*,v
```

```
echo “Working in the PCDS directory”
cd /u04/cm/lpswork/PCDS
get -vR2.0 -n $PVCS_LPS/PCDS/archives/.*,v
cd /u04/cm/lpswork/PCDS/data
get -vR2.0 -n $PVCS_LPS/PCDS/data/*,v
cd /u04/cm/lpswork/PCDS/db
get -vR2.0 -n $PVCS_LPS/PCDS/db/*,v
cd /u04/cm/lpswork/PCDS/db/include
get -vR2.0 -n $PVCS_LPS/PCDS/db/include/*,v
cd /u04/cm/lpswork/PCDS/db/src
get -vR2.0 -n $PVCS_LPS/PCDS/db/src/*,v
cd /u04/cm/lpswork/PCDS/global/include
get -vR2.0 -n $PVCS_LPS/PCDS/global/include/*,v
cd /u04/cm/lpswork/PCDS/global/src
get -vR2.0 -n $PVCS_LPS/PCDS/global/src/*,v
cd /u04/cm/lpswork/PCDS/include
get -vR2.0 -n $PVCS_LPS/PCDS/include/*,v
cd /u04/cm/lpswork/PCDS/scripts
get -vR2.0 -n $PVCS_LPS/PCDS/scripts/*,v
cd /u04/cm/lpswork/PCDS/src
get -vR2.0 -n $PVCS_LPS/PCDS/src/*,v

Echo “Working in the RDCS directory”
cd /u04/cm/lpswork/RDCS
get -vR2.0 -n $PVCS_LPS/RDCS/archives/.*,v
cd /u04/cm/lpswork/RDCS/data
get -vR2.0 -n $PVCS_LPS/RDCS/data/*,v
cd /u04/cm/lpswork/RDCS/bin
get -vR2.0 -n $PVCS_LPS/RDCS/bin/*,v
cd /u04/cm/lpswork/RDCS/db/src
get -vR2.0 -n $PVCS_LPS/RDCS/db/src/*,v
cd /u04/cm/lpswork/RDCS/global/include
get -vR2.0 -n $PVCS_LPS/RDCS/global/include/*,v
cd /u04/cm/lpswork/RDCS/global/src
get -vR2.0 -n $PVCS_LPS/RDCS/global/src/*,v
cd /u04/cm/lpswork/RDCS/include
get -vR2.0 -n $PVCS_LPS/RDCS/include/*,v
cd /u04/cm/lpswork/RDCS/scripts
get -vR2.0 -n $PVCS_LPS/RDCS/scripts/*,v
cd /u04/cm/lpswork/RDCS/src
get -vR2.0 -n $PVCS_LPS/RDCS/src/*,v
cd /u04/cm/lpswork/RDCS/src/src1
get -vR2.0 -n $PVCS_LPS/RDCS/src/src1/*,v
cd /u04/cm/lpswork/RDCS/src/src2
get -vR2.0 -n $PVCS_LPS/RDCS/src/src2/*,v
cd /u04/cm/lpswork/RDCS/src/src3
get -vR2.0 -n $PVCS_LPS/RDCS/src/src3/*,v
cd /u04/cm/lpswork/RDCS/src/src4
get -vR2.0 -n $PVCS_LPS/RDCS/src/src4/*,v
cd /u04/cm/lpswork/RDCS/src/src5
get -vR2.0 -n $PVCS_LPS/RDCS/src/src5/*,v
cd /u04/cm/lpswork/RDCS/src/src6
get -vR2.0 -n $PVCS_LPS/RDCS/src/src6/*,v
cd /u04/cm/lpswork/RDCS/src/src7
get -vR2.0 -n $PVCS_LPS/RDCS/src/src7/*,v
cd /u04/cm/lpswork/RDCS/src/src8
get -vR2.0 -n $PVCS_LPS/RDCS/src/src8/*,v
cd /u04/cm/lpswork/RDCS/src/src9
get -vR2.0 -n $PVCS_LPS/RDCS/src/src9/*,v

echo “Working in the RDPS directory”
cd /u04/cm/lpswork/RDPS
get -vR2.0 -n $PVCS_LPS/RDPS/archives/.*,v
cd /u04/cm/lpswork/RDPS/data
```

```
get -vR2.0 -n $PVCS_LPS/RDPS/data/*,v
cd /u04/cm/lpswork/RDPS/db/src
get -vR2.0 -n $PVCS_LPS/RDPS/db/src/*,v
cd /u04/cm/lpswork/RDPS/global/include
get -vR2.0 -n $PVCS_LPS/RDPS/global/include/*,v
cd /u04/cm/lpswork/RDPS/global/src
get -vR2.0 -n $PVCS_LPS/RDPS/global/src/*,v
cd /u04/cm/lpswork/RDPS/include
get -vR2.0 -n $PVCS_LPS/RDPS/include/*,v
cd /u04/cm/lpswork/RDPS/src
get -vR2.0 -n $PVCS_LPS/RDPS/src/*,v
cd /u04/cm/lpswork/RDPS/scripts
get -vR2.0 -n $PVCS_LPS/RDPS/scripts/*,v

echo "Working in the lpswork directory"
cd /u04/cm/lpswork
get -vR2.0 -n $PVCS_LPS/archives/.*,v
#get -vR2.0 -n $PVCS_LPS/archives/*,v

#scripts that extract from RCS (e.g. tco)
cd /u04/cm/lpswork/RCS_scripts
get -vR2.0 -n $PVCS_LPS/archives/*.tco,v
get -vR2.0 -n $PVCS_LPS/archives/makedir*,v
get -vR2.0  -n $PVCS_LPS/archives/extract*,v

echo "Working in the data directory"
cd /u04/cm/lpswork/data
get -vR2.0 -n $PVCS_LPS/data/*,v

echo "Working in the db directory"
cd /u04/cm/lpswork/db/include
get -vR2.0 -n $PVCS_LPS/db/include/*,v
cd /u04/cm/lpswork/db/src
get -vR2.0 -n $PVCS_LPS/db/src/*,v

echo "Working in the global directory"
cd /u04/cm/lpswork/global/include
get -vR2.0 -n $PVCS_LPS/global/include/*,v
cd /u04/cm/lpswork/global/src
get -vR2.0 -n $PVCS_LPS/global/src/*,v

echo "Working in the scripts directory"
cd /u04/cm/lpswork/scripts
get -vR2.0 -n $PVCS_LPS/scripts/*,v

echo "Working in the tables directory"
cd /u04/cm/lpswork/tables
get -vR2.0 -n $PVCS_LPS/tables/*,v

echo "Working in the ui directory"
cd /u04/cm/lpswork/ui/src
get -vR2.0 -n $PVCS_LPS/ui/src/*,v
```

C.3 LPS_makedir

```
#************************************************
#LPS_makedir
#
#Changes:
#  4/28/97 – added RCS_scripts (where the scripts used to extract units
#    from RCS are stored until they are checked into the pvcs/LPS/archives.
#  5/12/97 – additions to the tree for b3
#  5/29/97 – added new directories for b3: IDPS/src/src1-4, LDTS/src/src4,
```

```
#      MACS/src/src5-10
#  6/20/97 – added for b3.2: MACSsrc/src11, RDCS/src/src1..src9, v#B3.2
#  7/5/97 – problem – LPS_makedir loops?? hdf had too many dir levels
# 7/13/97 – changed v# to B3.2
# 7/15/97 – changed $INIT_HOME to $CM_HOME & -vB3.2 to -gST
#7/17/97 – changed -gST to -vR2.0 and checked into PVCS
#*****************************************************************
#!/bin/sh

echo "Making directory for LPS, the directory is called lpswork"
#Must go to HOME, then get lmake from pvcs
cd $CM_HOME
get -vR2.0 /usr/pvcs/LPS/archives/lmake,v
get -vR2.0 /usr/pvcs/LPS/archives/install,v

#make the home directory
cd $HOME
mkdir lpswork
cd lpswork
        mkdir COTS
        mkdir COTS/frame_sync
        mkdir COTS/frame_sync/include
        mkdir COTS/frame_sync/src
                cp $CM_HOME/lmake COTS/frame_sync/src/.
        mkdir COTS/frame_sync/obj
        mkdir COTS/frame_sync/scripts
                cp $CM_HOME/install COTS/frame_sync/scripts/.

                ln -s $CM_HOME/hdf COTS/hdf

        mkdir DAN

        mkdir DDN

        mkdir IDPS
        mkdir IDPS/bin
        mkdir IDPS/data
        mkdir IDPS/db
        mkdir IDPS/db/include
        mkdir IDPS/db/obj
        mkdir IDPS/db/src
                cp $CM_HOME/lmake IDPS/db/src/.
        mkdir IDPS/global
        mkdir IDPS/global/include
        mkdir IDPS/global/obj
        mkdir IDPS/global/src
                cp $CM_HOME/lmake IDPS/global/src/.
        mkdir IDPS/include
        mkdir IDPS/obj
        mkdir IDPS/scripts
         cp $CM_HOME/install IDPS/scripts/.
        mkdir IDPS/src
                cp $CM_HOME/lmake IDPS/src/.
        mkdir IDPS/src/src1
                 cp $CM_HOME/lmake IDPS/src/src1/.
        mkdir IDPS/src/src2
                cp $CM_HOME/lmake IDPS/src/src2/.
        mkdir IDPS/src/src3
                cp $CM_HOME/lmake IDPS/src/src3/.
        mkdir IDPS/src/src4
                cp $CM_HOME/lmake IDPS/src/src4/.

        mkdir LDTS
        mkdir LDTS/bin
```

```
mkdir LDTS/data
mkdir LDTS/db
mkdir LDTS/db/include
mkdir LDTS/db/obj
mkdir LDTS/db/src
        cp $CM_HOME/lmake  LDTS/db/src/.
mkdir LDTS/global
mkdir LDTS/global/include
mkdir LDTS/global/obj
mkdir LDTS/global/src
        cp $CM_HOME/lmake  LDTS/global/src/.
mkdir LDTS/include
mkdir LDTS/obj
mkdir LDTS/scripts
        cp $CM_HOME/install LDTS/scripts/.
mkdir LDTS/src
        cp $CM_HOME/lmake LDTS/src/.
mkdir LDTS/src/src1
        cp $CM_HOME/lmake LDTS/src/src1/.
mkdir LDTS/src/src2
        cp $CM_HOME/lmake LDTS/src/src2/.
mkdir LDTS/src/src3
        cp $CM_HOME/lmake LDTS/src/src3/.
mkdir LDTS/src/src4
        cp $CM_HOME/lmake LDTS/src/src4/.

mkdir MACS
mkdir MACS/bin
mkdir MACS/data
mkdir MACS/db
mkdir MACS/db/include
mkdir MACS/db/obj
mkdir MACS/db/src
        cp $CM_HOME/lmake MACS/db/src/.
mkdir MACS/global
mkdir MACS/global/include
mkdir MACS/global/obj
mkdir MACS/global/src
        cp $CM_HOME/lmake MACS/global/src/.
mkdir MACS/include
mkdir MACS/obj
mkdir MACS/scripts
        cp $CM_HOME/install MACS/scripts/.
mkdir MACS/src
        cp $CM_HOME/lmake MACS/src/.
mkdir MACS/src/src1
        cp $CM_HOME/lmake MACS/src/src1/.
mkdir MACS/src/src2
        cp $CM_HOME/lmake MACS/src/src2/.
mkdir MACS/src/src3
        cp $CM_HOME/lmake MACS/src/src3/.
mkdir MACS/src/src4
        cp $CM_HOME/lmake MACS/src/src4/.
mkdir MACS/src/src5
        cp $CM_HOME/lmake MACS/src/src5/.
mkdir MACS/src/src6
        cp $CM_HOME/lmake MACS/src/src6/.
mkdir MACS/src/src7
        cp $CM_HOME/lmake MACS/src/src7/.
mkdir MACS/src/src8
        cp $CM_HOME/lmake MACS/src/src8/.
mkdir MACS/src/src9
        cp $CM_HOME/lmake MACS/src/src9/.
mkdir MACS/src/src10
```

```
        cp $CM_HOME/lmake MACS/src/src10/.
mkdir MACS/src/src11
        cp $CM_HOME/lmake MACS/src/src11/.

mkdir MFPS
mkdir MFPS/bin
mkdir MFPS/data
mkdir MFPS/db
mkdir MFPS/db/include
mkdir MFPS/db/obj
mkdir MFPS/db/src
        cp $CM_HOME/lmake MFPS/db/src/.
mkdir MFPS/global
mkdir MFPS/global/include
mkdir MFPS/global/obj
mkdir MFPS/global/src
        cp $CM_HOME/lmake MFPS/global/src/.
mkdir MFPS/include
mkdir MFPS/obj
mkdir MFPS/scripts
        cp $CM_HOME/install MFPS/scripts/.
mkdir MFPS/src
        cp $CM_HOME/lmake MFPS/src/.

mkdir PCDS
mkdir PCDS/bin
mkdir PCDS/data
mkdir PCDS/db
mkdir PCDS/db/include
mkdir PCDS/db/obj
mkdir PCDS/db/src
        cp $CM_HOME/lmake PCDS/db/src/.
mkdir PCDS/global
mkdir PCDS/global/include
mkdir PCDS/global/obj
mkdir PCDS/global/src
        cp $CM_HOME/lmake PCDS/global/src/.
mkdir PCDS/include
mkdir PCDS/obj
mkdir PCDS/scripts
        cp $CM_HOME/install PCDS/scripts/.
mkdir PCDS/src
        cp $CM_HOME/lmake PCDS/src/.

mkdir RDCS
mkdir RDCS/bin
mkdir RDCS/data
mkdir RDCS/db
mkdir RDCS/db/include
mkdir RDCS/db/obj
mkdir RDCS/db/src
        cp $CM_HOME/lmake RDCS/db/src/.
mkdir RDCS/global
mkdir RDCS/global/include
mkdir RDCS/global/obj
mkdir RDCS/global/src
        cp $CM_HOME/lmake RDCS/global/src/.
mkdir RDCS/include
mkdir RDCS/obj
mkdir RDCS/scripts
        cp $CM_HOME/install RDCS/scripts/.
mkdir RDCS/src
        cp $CM_HOME/lmake RDCS/src/.
mkdir RDCS/src/src1
```

```
            cp $CM_HOME/lmake RDCS/src/src1/.
mkdir RDCS/src/src2
            cp $CM_HOME/lmake RDCS/src/src2/.
mkdir RDCS/src/src3
            cp $CM_HOME/lmake RDCS/src/src3/.
mkdir RDCS/src/src4
            cp $CM_HOME/lmake RDCS/src/src4/.
mkdir RDCS/src/src5
            cp $CM_HOME/lmake RDCS/src/src5/.
mkdir RDCS/src/src6
            cp $CM_HOME/lmake RDCS/src/src6/.
mkdir RDCS/src/src7
            cp $CM_HOME/lmake RDCS/src/src7/.
mkdir RDCS/src/src8
            cp $CM_HOME/lmake RDCS/src/src8/.
mkdir RDCS/src/src9
            cp $CM_HOME/lmake RDCS/src/src9/.

mkdir RDPS
mkdir RDPS/bin
mkdir RDPS/data
mkdir RDPS/db
mkdir RDPS/db/include
mkdir RDPS/db/obj
mkdir RDPS/db/src
      cp $CM_HOME/lmake RDPS/db/src/.
mkdir RDPS/global
mkdir RDPS/global/include
mkdir RDPS/global/obj
mkdir RDPS/global/src
      cp $CM_HOME/lmake RDPS/global/src/.
mkdir RDPS/include
mkdir RDPS/obj
mkdir RDPS/scripts
            cp $CM_HOME/install RDPS/scripts/.
mkdir RDPS/src
            cp $CM_HOME/lmake RDPS/src/.

mkdir bin

mkdir data

mkdir db
mkdir db/include
mkdir db/obj
mkdir db/src
            cp $CM_HOME/lmake db/src/.

mkdir global
mkdir global/include
mkdir global/obj
mkdir global/src
            cp $CM_HOME/lmake global/src/.

mkdir journal

mkdir man

mkdir reports

mkdir RCS_scripts

mkdir scripts
```

```
        mkdir tables

        mkdir tools
        mkdir tools/bin
        mkdir tools/include
        mkdir tools/obj
        mkdir tools/src
                cp $CM_HOME/lmake tools/src/.

        mkdir troublefile

        mkdir ui
        mkdir ui/bin
        mkdir ui/include
        mkdir ui/obj
        mkdir ui/src
                cp $CM_HOME/lmake ui/src/.

#setenv CM_HOME $HOME/lpswork
echo "lpswork directory built"
```

C.4 setup_st_deliv

This script can be used to set up an "at" directory also.

```
#set_st_deliv : is to set up the structure for st that will be ftp'd to st machine
# Process:
# change the b# in the script before proceeding "mkdir b##_4_st" and "cd b#_4_st"
# 7/6/97 – added notes and cp bin to dir.
# 7/14/97 – added mkdir schedules and iasparms
# 7/31/97 – cleaned up for R2.0; changes expected for 2.1
#************
#*Changes
#
#
cd $HOME/Deliv_4_st/b32p1_delst
mkdir b32p1_4_st
cd b32p1_4_st
mkdir DAN
mkdir DDN
mkdir troublefile
mkdir reports
mkdir schedules
mkdir iasparms

#bin & tables dirs will be made when copying the bin and tables from lpswork
      cp -pr $LPS_HOME/bin.
      cp -pr $LPS_HOME/tables.

cp -p $LPS_HOME/.lpsrc.

cp -p $LPS_HOME/IDPS/src/src4/Idp_mwd .

cp -p $ORACLE_HOME/guicommon2/tk21/admin/Tk2Motif.gray .

mv Tk2Motif.gray Tk2Motif
echo "See Build Instructions to edit TK2Motif:"
echo "Edit Tk2Motif: find the word 'Private'"
echo " – see Build Instr. page 13 of Attach D."
echo "Have System Administrator change ownership to root of"
echo "rdc_Capture, rdc_Transmit, rdc_DeleteFiles"
echo "and to set user ID: chmod u+s <each file>."
echo "If debug build is complete, copy the bin into the st deliv as d_bin"
```

Appendix D. Sample Environment Variables for System Test

Source .lpsrc to verify the environment for system test.

HOME=/export/home/lpsst

LOGNAME=lpsst

HZ=100

TZ=EST5EDT

TERM=iris-ansi-net

USER=lpsst

LANG=C

SHELL=/bin/tcsh

REMOTEHOST=lpsdev1.gsfc.nasa.gov

REMOTEUSER=UNKNOWN

MAIL=/usr/mail/lpsst

DISPLAY=lpscm.gsfc.nasa.gov:0.0

SHLVL=1

PWD=/export/home/lpsst

HOST=lps003

HOSTTYPE=iris4d

MSGVERB=text:action

NOMSGLABEL=1

NOMSGSEVERITY=1

LD_LIBRARYN32_PATH=/usr/lib32

LD_LIBRARY64_PATH=/usr/lib64

LPS_HOME=/usr/LPS/st/b3.2

HDF_HOME=/u01/hdf/HDF3.3r4/hdf

LPS_LOG_STDOUT=TRUE

PATH=.:/usr/home/lpsst/bin:/export/home/lpsst/tools:/usr/LPS/st/b3.2/bin/:/usr/app/oracle/client/developer2000/1.3.1/bin:/usr/bin/X11/:/usr/local/bin:/usr/bin:/bin:/usr/etc:/etc:/usr/bsd:/usr/sbin:/bin

MANPATH=/usr/LPS/st/b3.2/man:/usr/share/catman:/usr/share/man:/usr/catman:/usr/local/man

LD_LIBRARY_PATH=/usr/app/oracle/client/developer2000/1.3.1/lib:/usr/lib

X11HOME=/usr/bin/X11

XNLSPATH=/usr/motif/lib/X11/nls

ORACLE_HOME=/usr/app/oracle/client/developer2000/1.3.1

ORACLE_PATH=/usr/app/oracle/client/developer2000/1.3.1/bin:/usr/bin/X11/:/usr/local/bin:/usr/bin:/bin:/usr/etc:/etc:/usr/bsd:/usr/sbin:/sbin

ORACLE_SID=LPS

ORACLE_TERM=xterm

TNS_ADMIN=/usr/app/oracle/client/developer2000/1.3.1/tns

TWO_TASK=lps

LPS_BIN=/usr/LPS/st/b3.2/bin

LPS_DANFILE_PATH=/usr/LPS/st/b3.2/DAN

LPS_DDNFILE_PATH=/usr/LPS/st/b3.2/DDN

LPS_JOURNAL_PATH=/usr/LPS/

LPS_OUTFILE_PATH=/u02/st/b3.2/outfile

LPS_RAWFILE_PATH=/u01/st/b3.2/rawfile

LPS_REPORT_PATH=/usr/LPS/st/b3.2/reports

LPS_TAPE_DEV=/dev/rmt/tps131d2vc

LPS_TABLE_PATH=/usr/LPS/st/b3.2/tables

LPS_TEMPFILE_PATH=/u01/st/b3.2

LPS_TROUBLEFILE_PATH=/u02/st/b3.2/troublefile

LPS_IAS_PARMS_PATH=/usr/LPS/st/b3.2/iasparms

LPS_CONT_SCHED_PATH=/usr/LPS/st/b3.2/schedules

LPS_PRINTER_DEVICE=/dev/plp

LPS_TAPE_LIBRARY_DEV=/dev/scsi/sc131d2l0

RDC_DEVICE=/dev/hpdiB

RDC_STATUS_INTERVAL=30

RDC_THRESH_SYSTEMDISK=0.01

LPS_CAPTURE_PROCESSOR=1

```
ENV=/export/home/lpsst/.kshrc
```

Appendix E. Sample Tester's .cshrc File

```
###############################################################################
# Sample .cshrc for LPS users/developers                                      #
###############################################################################

# Set up for remote protocol with other servers
if ($?USER == 0 || $?prompt == 0) exit

setenv LPS_HOME /usr/LPS/at/r2
setenv HDF_HOME /u01/hdf/HDF3.3r4/hdf
setenv LPS_LOG_STDOUT TRUE

# Add LPS environment settings
if (-e $LPS_HOME) then
  # Force a clean path (recommended)
  unsetenv PATH
  unsetenv MANPATH
  unsetenv LD_LIBRARY_PATH

  # Set the default path and sets up LPS operations/test environment
  # Developers/maintenance programmers use .lpsdevrc instead of .lpsrc
#  source $LPS_HOME/.lpsdevrc
source $LPS_HOME/.lpsrc
#   source /usr/home/lpsst/.lpsrc
else
  echo "LPS_HOME is not available."
endif

# Add my own environment
setenv PATH .:/usr/home/lpsst/bin:/export/home/lpsst/tools:$PATH

# Modify default environment variables

# File completion using the escape but ignoring certain extensions.
set filec
set fignore=.o

# Set the history
set history=40

# Set the prompt
# set prompt="`uname -n`:`whoami`:\!:"
set prompt="`hostname`:%/[%!\] %"

# Set the backspace key (Ctrl-v then backspace)
if ($TERM == sun-cmd || $TERM == xterm) then
  stty echo erase
endif

# Initialize all aliases
if (-e $HOME/.alias) then
  source $HOME/.alias
endif
```

Appendix F. Installing the IRIX Operating System

Follow the vendor's installation instructions using the following checklist:

1. Repartition the system disk if it's not already partitioned.
   a. Select System Maintenance Menu, option 5:

      boot stand/fx --x

   b. Make filesystem (mkfs <filesystem name>) for the new partitions.
   c. Mount new filesystem.
2. Install IRIX 6.2 CDs (Parts 1 and 2).
3. Install ONC3/NFS 6.2.
4. Install IRIS Development Option 6.2.
5. Install IRIX Applications May 1996.
6. Install IRIS Power C 6.2.
7. Reboot and edit the following files to contain site-dependent information (i.e., ip address or nameserver):

   /etc/sys_id

   /etc/hosts

   /etc/passwd

   /etc/hosts

   /etc/resolv.conf

   /etc/TIMEZONE

   /etc/config/netif.options

   NOTE: /etc/config/netif.options should define the following interface:

   if1name=et0

   if1addr=$HOSTNAME

   if2name=xpi0

   if2addr=gate-$HOSTNAME

   if3name=xpi1

   if3addr=gate2-$HOSTNAME

8. Reboot again and verify network connectivity via ping or telnet.
9. Mount RAIDs as follows, if not already done:

   mkdir /u01

   mkdir /u02

   mount /dev/dsk/dks133d1s7 /u01

   mount /dev/dsk/dks134d1s7 /u02

   mount -p > /etc/fstab
10. Install IRIX 6.2 recommended patches:

    netscape http://www.surf.sgi.com/SurfZone/Support/patchset/index.html
11. Install HPDI/VSIO code.

Make the following modifications after installation is complete (you must be "root" to make them):

1. Switch user to root:

   su Root
2. Edit /etc/syslog.conf:
   a. Add the following line at the end of the file:

      local0.debug;kern.none /tmp/LPS_Journal

   b. Comment out:

      local0.alert root, operator

   c. Save the file.

   d. Issue the **killall -HUP syslogd** command.
3. Run systune interactively by typing **systune -i**:
   a. Change semmns to 2048

   b. Change semmni to 100

   c. Quit and reboot

Appendix G. Installing Network Time Protocol

To install Network Time Protocol on each string, follow these steps:

1. Download

   from IRIX 6.2 Freeware CD or SGI download site

   (http://www.sgi.com/TasteOfDT/public/freeware2.0/)

2. Follow the vendor's installation instructions as provided on the CD or at the Internet site.

3. The location, /usr/freeware, will be created during a successful installation.

Appendix H. Installing ORACLE

H.1 Installing the ORACLE Server

The ORACLE Server (V 7.3.2) was installed following builder's installation instructions. The following responses are entered at the installer prompts:

To move around the installer window, use the tab key to move from one block to another block. Use the arrow key to move from one field to another field. To make a selection, use the space bar.

1. Login to the oracle account (default shell of ORACLE is Bourne shell).
2. Verify that the oracle home directory is /usr/app/oracle by typing:

   pwd
3. Create the server directory:

   mkdir /usr/app/oracle/product; mkdir /usr/app/oracle/product/7.3.2;
4. Set the environment variables. Create .profile. The content of .profile should be as follows:

   ORACLE_SID=LPS

   export ORACLE_SID

   ORACLE_HOME=/usr/app/oracle/product/7.3.2

   export ORACLE_HOME

   ORACLE_TERM=iris

   export ORACLE_TERM

   LD_LIBRARY_PATH=$ORACLE_HOME/lib

   export LD_LIBRARY_PATH

   PATH=$ORACLE_HOME/bin:$PATH

   export PATH

   ORACLE_PATH=$PATH

   export ORACLE_PATH

   Execute the .profile by typing:

   . .profile
5. Load the CD into the CD-ROM driver.

6. Run the installer:

   cd /CDROM/orainst

   ./orainst

   Press OK at the Welcome to the New ORACLE Installer screen.

7. Enter the following information at the prompts:

   Select the Installer activity:

   Install, Upgrade, or De-Install Software

   Select the Installer option:

   Install New Product

   Enter mount point for your software installation: /usr

   OK

   Complete $ORACLE_HOME location: /usr/app/oracle/product/7.3.2

   OK

   Do you want to create DB objects also?

   Yes

   Confirm (or Change) Log File Location:

   Installer Log: log

   SQL Log: /usr/app/oracle/product/7.3.2/orainst/sql.log

   Makefile Log: /usr/app/oracle/product/7.3.2/orainst/make.log

   OS Log: /usr/app/oracle/product/7.3.2/orainst/os.log

   OK

   For the README file /CDROM/orainst/README.FIRST:

   OK

   Do not display this README file in the future

   OK

   Select one of the following:

   Install from CD-ROM

   Enter your ORACLE_SID: LPS

OK

Select the native language to be installed:

American/English

Do you want to relink ORACLE product executables?

Yes

Post-installation steps that need to be run by root will be written to /usr/app/oracle/product/7.3.2/orainst/root.sh.

OK

Install online help for:

All Products Being Installed

Do you want to install UNIX-specific documentation?

No

Install product documentation from the product documentation CD-ROM for:

No Products

Select the following products from the product list:

Advanced Replication Option 7.3.2.1.0

IRIX Specific Enhancements 7.3.2.1.0

ORACLE Intelligent Agent 7.3.2.1.0

ORACLE Names 2.0.1.1.0

ORACLE Server Manager (Motif)

ORACLE7 Distributed Database option 7.3.2.1.0

ORACLE7 Server (RDBMS) 7.3.2.1.0

ORACLE7 XA Library 1.1.1.0.0

PL/SQL V2 2.3.2.0.0

Pro*C 2.2.2.0.0

SQL*Module 1.1.4.0.0

SQL*Net (V2) 2.3.2.1.0

SQL*Plus 3.3.2.0.0

TCP/IP Protocol Adapter (V2) 2.3.2.1.0

Install

Enter the official hostname (including domain) for this server:

<host name>.<domain name>

(For example, in lps002.gsfc.nasa.gov, lps002 is the hostname and gsfc.nasa.gov is the domain name.)

OK

Enter a TCP service port for the Administration Server. 8888

OK

Enter password for the Web administration. Reenter the same password at the verifying prompt.

Select group to act as database administrator of the database:

dba

Select the OSOPER group: dba

OK

Choose storage type for database:

Filesystem-Based Database

Distribute control files over three mount points?

Yes

Enter /usr/app/oracle for all three mount points.

Select the character set for creating this database.

US7ASCII

Enter passwords for SYSTEM and SYS. Reenter the same password at the verifying prompt.

Do you want to set the passwords for the internal users (dba and operator)?

No

Enter the password for the TNS listener.

Would you like MTS (Multi-Threaded Server) configured and the SQL*Net listener automatically started?

No

Do you want to use the following files as control files?

/usr/app/oracle/oradata/LPS/control01.ctl

/usr/app/oracle/oradata/LPS/control02.ctl

/usr/app/oracle/oradata/LPS/control03.ctl

Yes

The following are the default file names and sizes that will be used to create the new database. Select (OK) to continue to the next screen of defaults files.

Default file names and sizes continued. Select (Back) to see the previous screen of defaults. Select (OK) to continue.

OK

Select (Yes) to accept the default file names and sizes shown on the previous screens. Select (No) to specify new values. Select (Back) to view the previous screens.

Yes

You've selected to install the IRIX Specific Enhancements....

OK

Would you like to load the SQL*Plus Help Facility?

Yes

Would you like to load the SQL*Plus Demo Tables?

Yes

Please enter the directory where the X-Windows libraries (libXt, libX11) may be found: /usr/lib

OK

Installation of shared oracle library for Pro*C, OCI, and XA clients is complete.

OK

The requested action has been performed for selected products. Select (OK) to continue.

OK

8. Exit from the installer

At this point, directories (admin, local, oradata) are created by the installer, and the database instance creation scripts exist in the create directories.

9. Run the root.sh script as the root.

   su root

   cd $ORACLE_HOME/orainst

   ./root.sh

   The following environment variables are set. Are these settings correct (Y/N)? [Y]: Y

   Enter the full pathname of the local bin directory [/opt/bin]: Return

   ORACLE_HOME does not match the home directory for oracle.

   Okay to continue? [N]: Y

   Exit from the root account and return to the oracle account by typing: exit

10. The ORACLE instance is created and up and running. To test login, type

    sqlplus

    Enter user-name: system

    Enter password: enter the password defined during the installation.

    SQL> exit

11. Unload the CD:

    cd (to get the installer off the /CDROM)

    eject

H.2 Upgrading the ORACLE Server

ORACLE server 7.3.2.2 is a patch software. It should be installed after installing 7.3.2.

1. Login to the oracle account.
2. To upgrade to a newer version, the database should be shut down:

   svrmgrl

   SVRMGR> connect internal;

   SVRMGR> shutdown;

   SVRMGR> quit
3. Run the installer:

   cd /CDROM/orainst

   ./orainst

   Press OK at the Welcome to the New ORACLE Installer screen.
4. Enter the following information at the prompts:

   Select the Installer activity:

   Install, Upgrade, or De-Install Software

   Select the Installer option (use the Tab key to move to the selection list. Use the arrow key to go to the Add/Upgrade Software selection. Use the space bar to select the option.):

   Add/Upgrade Software

   OK

   Enter $ORACLE_HOME location: /usr/app/oracle/product/7.3.2

   OK

   Confirm (or Change) Log File Location:

   Installer Log: log

   SQL Log: /usr/app/oracle/product/7.3.2/orainst/sql.log

   Makefile Log: /usr/app/oracle/product/7.3.2/orainst/make.log

   OS Log: /usr/app/oracle/product/7.3.2/orainst/os.log

   OK

For the README file /CDROM/orainst/README.FIRST :

OK

Do not display this README file in the future

OK

Select one of the following:

Install from CD-ROM

Select the native language to be installed:

American/English

OK

Relink All Executables?

Yes

The /usr/app/oracle/product/7.3.2/orainst/root.sh file already exists. Select one of the following actions:

Create new root.sh file

OK

Enter the pathname where the existing root.sh file will be saved: /usr/app/oracle/product/7.3.2/orainst/root.sh0

OK

Post-installation steps that need to be run by root will be written to /usr/app/oracle/product/7.3.2/orainst/root.sh.

OK

Install online help for:

All Products Being Installed

OK

Do you want to install UNIX-specific documentation?

No

Install product documentation from the product documentation CD-ROM for:

No Products

OK

Select the following products from the product list:

Advanced Replication Option 7.3.2.2.0

IRIX Specific Enhancements 7.3.2.2.0

ORACLE Context Option 1.1.2.0.0

ORACLE UNIX Installer and Documentation Viewer

ORACLE7 Distributed Database option 7.3.2.2.0

ORACLE7 Server (RDBMS) 7.3.2.2.0

ORACLE7 XA Library 7.3.2.2.0

PL/SQL V2 2.3.2.2.0

Spatial Data Option 7.3.2.2.0

Install

The currently installed version of ORACLE Common Libraries and Utilities software, 7.3.2.1.0, is out of date. The most current version is 7.3.2.2.0. Do you want to upgrade the software to 7.3.2.2.0?

Yes

The currently installed version of PL/SQL V2 software, 2.3.2.0.0, is out of date. The most current version is 2.3.2.2.0. Do you want to upgrade the software to 2.3.2.2.0?

Yes

The currently installed version of ORACLE7 Distributed Database option software, 7.3.2.1.0, is out of date. The most current version is 7.3.2.2.0. Do you want to upgrade the software to 7.3.2.2.0?

Yes

The currently installed version of Advanced Replication Option software, 7.3.2.1.0, is out of date. The most current version is 7.3.2.2.0. Do you want to upgrade the software to 7.3.2.2.0?

Yes

The currently installed version of ORACLE7 Server (RDBMS) software, 7.3.2.1.0, is out of date. The most current version is 7.3.2.2.0. Do you want to upgrade the software to 7.3.2.2.0?

Yes

Select group to act as DBA of the database:

dba

OK

Select the OSOPER group:

dba

OK

Enter your ORACLE_SID:

LPS

OK

Does the LPS database exist and is it functional?

No

The currently installed version of IRIX Specific Enhancements software, 7.3.2.1.0, is out of date. The most current version is 7.3.2.2.0. Do you want to upgrade the software to 7.3.2.2.0?

Yes

You've selected to install the IRIX Specific Enhancements. Before using them, verify that you have the database accelerator (dba), Generic Database Accelerator Features (eoe.sw.database_accel) and patches 1167 (patchSG0001167) and 1194 (patchSG0001194) installed on your system by running the /usr/sbin/versions command.

OK

Please enter the password for the 'MDSYS' ORACLE userid:

enter the password

OK

The currently installed version of ORACLE7 XA Library software, 1.1.1.0.0, is out of date. The most current version is 7.3.2.2.0. Do you want to upgrade the software to 7.3.2.2.0?

OK

The currently installed version of SLAX: parser software, 7.3.2.1.0, is out of date. The most current version is 7.3.2.2.0. Do you want to upgrade the software to 7.3.2.2.0?

Yes

Would you like to install ORACLE Context Option demos?

Yes

Would you like to regenerate shared version of the ORACLE library for Pro*C, OCI, and XA clients?

Yes

Select (Help) for more details on what you can do next.

Select (OK) to continue.

OK

Exit from the installer.

Exit

5. Deinstall IRIX Specific Enhancements:

cd /CDROM/orainst

./orainst

Press OK at the Welcome to the New ORACLE Installer screen.

Select the Installer activity:

Install, Upgrade, or De-Install Software

OK

Select the Installer option:

De-Install Software

OK

Enter $ORACLE_HOME location:

/usr/app/oracle/product/7.3.2

OK

Confirm (or Change) Log File Location:

OK

Select IRIX Specific Enhancements 7.3.2.2.0

Remove

Installer Actions Completed

Select (OK) to continue.

OK

Exit

6. Unload the CD

   cd (to get the installer off the /CDROM)

   eject

H.3 Creating a Database Instance

1. In the oracle account

   Check if the ORACLE instance, LPS is running:

   ps -ef | grep ora

   (should not return ora_smon_LPS, ora_pmon_LPS, ora_dbwr_LPS, ora_lgwr_LPS, and ora_reco_LPS)

2. Modify crdbLPS.sql to change the size of the database.

   cd /usr/app/oracle/admin/LPS/create

   '/usr/app/oracle/oradata/LPS/system01.dbf' size 50M

   '/usr/app/oracle/oradata/LPS/redoLPS01.log' size 2M,

   '/usr/app/oracle/oradata/LPS/redoLPS02.log' size 2M,

   '/usr/app/oracle/oradata/LPS/redoLPS03.log' size 2M;

3. Modify crdb2LPS.sql to change the size of the database.

   Add spool file creation statement under the #spool 2-rdbms.lst:

   spool /usr/app/oracle/admin/LPS/create/crdb_LPS_second.lst

   Modify sizes:

   create tablespace temp datafile

   '/usr/app/oracle/oradata/LPS/temp01.dbf' size 5M

   default storage (

   initial 1m

   next 1m

   pctincrease 0

   );

   '/usr/app/oracle/oradata/LPS/users01.dbf' size 30M;

   Create two more tablespaces. Add the following statements after creation of users tablespace:

   REM * Create two tablespaces for application

   REM *

create tablespace appl_data datafile

'/usr/app/oracle/oradata/LPS/appl_data.dbf' size 30M

default storage(

initial 10K

next 10K

pctincrease 0

minextents 1

);

create tablespace appl_index datafile

'/usr/app/oracle/oradata/LPS/appl_index.dbf' size 15M

default storage(

initial 10K

next 10K

minextents 1

pctincrease 0

);

Add the following script executing command before executing catdbsyn.sql:

connect internal

@/usr/app/oracle/product/7.3.2/rdbms/admin/catproc.sql

Add the following script executing command after executing catdbsyn.sql:

connect system/manager

@/usr/app/oracle/product/7.3.2/sqlplus/admin/pupbld.sql

4. Modify initLPS.ora and initLPS_0.ora to the following values:

cd /usr/app/oracle/admin/LPS/pfile

db_files = 60

#Since advanced replication option was installed

shared_pool_size = 10000000

#shared_pool_size = 3500000

global_names = TRUE

Add the following parameters:

remote_os_authent = true

open_links = 20

5. Create the instance by executing the creation scripts.

   cd /usr/app/oracle/admin/LPS/create

   svrmgrl

   SVRMGR> connect internal

   SVRMGR>@crdbLPS.sql

   SVRMGR>@crdb2LPS.sql

   SVRMGR> quit

6. Check the instance is up and running.

   ps -ef | grep ora

   (should return these background processes: ora_reco_LPS, ora_pmon_LPS, ora_lgwr_LPS, ora_dbwr_LPS, and ora_smon_LPS)

7. Verify login.

   sqlplus

   Enter user-name: system

   Enter password: manager (default password)

   SQL> alter user system identified by xxxxx;

   SQL> alter user sys identified by xxxxx;

   SQL> quit

H.4 Creating the LPS Logical Database

1. Login to the oracle account.
2. Make a new directory and copy logical database building scripts listed in db_creation_notes to the new directory.

   mkdir /usr/app/oracle/admin/LPS/create/logi_db_creation

   cd /usr/app/oracle/admin/LPS/create/logi_db_creation

   cp $LPS_TABLE_PATH/tables/* .
3. Make sure that ORACLE_HOME and ORACLE_SID are set correctly.

   echo $ORACLE_HOME

   /usr/app/oracle/product/7.3.2

   echo $ORACLE_SID

   LPS
4. If the database instance is not running, use the following procedures to start it:

   svrmgrl

   SVRMGR> connect internal

   SVRMGR> startup;

   SVRMGR> quit

   At this point, no user exists.
5. The following steps are stated in the db_creation_notes script with a minor difference.
6. Logon as system into sqlplus.

   sqlplus system/password
7. Create and grant quotas for the application account owner (appldba) and grant connect role to account owner.

   SQL> @create_appldba_user.sql

   SQL> select * from all_users;

   SQL> connect appldba

   SQL> select * from session_privs;

   SQL> connect system

8. Create the role(s)

   SQL> @create_role.sql

   SQL> select * from dba_roles;

9. Create all other users

   SQL> @create_lpsat_user.sql

   SQL> connect OPS$LPSAT/LPSAT

   SQL> select * from session_privs;

   SQL> connect system

10. Grant the role(s) to users

    SQL> @grant_role.sql

    SQL> select * from dba_role_privs;

11. Connect as the account owner (appldba)

    SQL> connect appldba

12. Create the tables and sequences

    SQL> @create_table.sql

    SQL> select * from tab;

    SQL> @create_sequence.sql

    SQL> select * from seq;

13. Generate the table and sequence privileges grants creation script

    SQL> @grant_privs_to_roles.sql

14. Generate the public synonyms creation script

    SQL> @create_public_syn.sql

15. Grant the table and sequence privileges to the roles

    SQL> @tmp_grants.sql

    SQL>connect system

    SQL>set linesize 120

    SQL>set pages 100

    SQL>column privilege format a10

SQL>select TABLE_NAME,PRIVILEGE,GRANTEE,OWNER from dba_tab_privs

> where GRANTOR='APPLDBA'

> order by TABLE_NAME,PRIVILEGE;

16. Connect as system and create the public synonyms

    SQL>connect system

    SQL> @tmp_pubsyn.sql

    SQL>select SYNONYM_NAME , TABLE_NAME from dba_synonyms

    > where OWNER = 'PUBLIC' and TABLE_OWNER = 'APPLDBA';

    SQL>connect ops$lpsat

    SQL>desc PCD_MJF_ACCT;

17. Connect as account owner (appldba) and load the static data

    SQL> @load_static_data.sql

    SQL>select * from LPS_Configuration;

    SQL>select * from Valid_CCSDS_Parms;

    SQL>select * from valid_ldt_parms;

    SQL>select * from Valid_RDP_Thres;

18. Modify LPS_Configuration:

    SQL>update LPS_Configuration set LPS_HARDWARE_STRING_ID = 'lps002';

19. Log out and load the static data for valid_wrs_parms:

    pcd_sqlldr_command

    (this will use sql loader and data file pcd_data_file.dat, control file pcd.ctl and writes to the log pcd.log)

    Login sqlplus check the static table data:

    sqlplus appldba

    SQL> select count(*) from valid_wrs_parms;

20. Login as system and create database link:

    sqlplus system

    SQL>@create_db_link.sql

SQL>select * from dba_db_links;

SQL> quit

21. Delete scripts by removing the /usr/app/oracle/admin/LPS/create/logi_db_creation directory:

    rm -r /usr/app/oracle/admin/LPS/create/logi_db_creation

H.5 Installing the Developer 2000

1. Login to the oracle account.
2. Before running the installer, create the client and tns admin directories: /usr/app/oracle/client/developer2000/1.3.1 and /usr/app/oracle/client/developer2000/1.3.1/tns

   mkdir client; cd client; mkdir developer2000; cd developer2000;

   mkdir 1.3.1; cd 1.3.1; mkdir tns;

3. Set the environment variables as follows:

   ORACLE_HOME=/usr/app/oracle/client/developer2000/1.3.1

   export ORACLE_HOME

   ORACLE_SID=LPS

   export ORACLE_SID

   TNS_ADMIN=$ORACLE_HOME/tns

   export TNS_ADMIN

   ORACLE_TERM=iris

   export ORACLE_TERM

   LD_LIBRARY_PATH=$ORACLE_HOME/lib

   export LD_LIBRARY_PATH

   PATH=$ORACLE_HOME/bin:$PATH

   export PATH

   ORACLE_PATH=$PATH

   export ORACLE_PATH

4. Load the CD and run the installer.

   cd /CDROM/orainst

./orainst

5. Enter the following information at the prompts:

At the installer welcome window prompt, press OK.

Select the Installer activity:

Install, Upgrade, or De-Install Software

OK

Select the Installer option:

Add/Upgrade Software

OK

Enter $ORACLE_HOME location:

/usr/app/oracle/client/developer2000/1.3.1

OK

Confirm (or Change) Log File Location:

SQL Log: /usr/app/oracle/client/developer2000/1.3.1/orainst

Makefile Log: /usr/app/oracle/client/developer2000/1.3.1/orainst

OS Log: /usr/app/oracle/client/developer2000/1.3.1/orainst

OK

For the README file /CDROM/orainst/README.FIRST :

Do not display this README file in the future

OK

Select one of the following:

Install from CD-ROM

OK

Select the native language to be installed:

American/English

OK

Do you want to relink ORACLE product executables?

Yes

Post-installation steps that need to be run by root will be written to /usr/app/oracle/client/developer2000/1.3.1/orainst/root.sh

OK

Install online help for:

All Products Being Installed

Do you want to install UNIX-specific documentation?

No

Install product documentation from the product documentation CD-ROM for:

No Products

OK

Select the following products from the list:

ORACLE Forms 4.5.7.0.12

ORACLE Reports 2.5.5.1.1

PL/SQL Procedure Builder 1.5.6.12.1

SQL*Net (V2) 2.3.2.1.0

SQL*Plus 3.3.2.0.0

TCP/IP Protocol Adapter (V2) 2.3.2.1.0

Install

Would you like to install the PL/SQL Procedure Builder demos?

Yes

Please enter the directory where the X-Windows libraries (libXt, libX11) may be found: /usr/lib

OK

Please enter the directory where the MOTIF library (libXm) is located: /usr/lib

OK

Would you like to install the ORACLE Graphics 2.5 demos?

No

Select one or more user interfaces for ORACLE Forms...

Motif Bitmapped Interface

OK

Would you like to install the ORACLE Forms demos?

No

Select one or more user interfaces for ORACLE Reports 2.5...

Motif Bitmapped Interface

OK

Would you like to install the ORACLE Reports 2.5 demos?

No

Installation of shared oracle library for Pro*C, OCI, and XA clients is complete.

OK

The requested action has been performed for selected products. Select (OK) to continue.

OK

6. Exit from the installer

   OK

7. Run the root.sh script as the root.

   su root

   cd $ORACLE_HOME/orainst

   ./root.sh

   The following environment variables are set as:

   ORACLE_OWNER= oracle

   ORACLE_HOME= /usr/app/oracle/client/developer2000/1.3.1

   ORACLE_SID= LPS

   Are these settings correct (Y/N)? [Y]:

   Enter the full pathname of the local bin directory [/opt/bin]:

   ORACLE_HOME does not match the home directory for oracle.

   Okay to continue? [N]: Y

   exit

8. Unload the CD.

   cd (to get the installer off the /CDROM)

   eject

9. Copy tnsnames.ora into $TNS_ADMIN directory from LPS_TABLE_PATH/tables.

   cp $LPS_TABLE_PATH/tables/tnsnames.ora $TNS_ADMIN

10. Copy listener.ora into /usr/app/oracle/product/7.3.2/network/admin cp $LPS_TABLE_PATH/tables/listener.ora /usr/app/oracle/product/7.3.2/network/admin

    Modify listener.ora to change the host ip address to proper one.

11. Add the following into .profile

    TNS_ADMIN=/usr/app/oracle/client/developer2000/1.3.1/tns

    export

12. Login to the oracle account. ORACLE_HOME should point to the server directory:

    echo $ORACLE_HOME

    /usr/app/oracle/product/7.3.2

13. Bring up listener:

    lsnrctl start

    LSNRCTL> quit

14. Bring up the database instance:

    svrmgrl

    SVRMGR> connect internal;

    SVRMGR> startup

    SVRMGR> quit

15. To bring up the forms/report from any user account:

    Login to an account.

    Set TWO_TASK to the proper connect alias defined in tnsnames.ora

    TWO_TASK=LPS2_LPS

    export TWO_TASK

    The following environment variables should be set.

```
ORACLE_HOME=/usr/app/oracle/client/developer2000/1.3.1
export ORACLE_HOME
TNS_ADMIN=$ORACLE_HOME/tns
export TNS_ADMIN
ORACLE_TERM=xterm
export ORACLE_TERM
LD_LIBRARY_PATH=$ORACLE_HOME/lib
export LD_LIBRARY_PATH
PATH=$ORACLE_HOME/bin:$PATH
export PATH
ORACLE_PATH=$PATH
export ORACLE_PATH
```

Run the designer to test:

```
f45desm &
```

Appendix I. Sample Environment Variables for CM

Note: The paths should be the same with the exception of /u04.

HOME=/u04/cm

LOGNAME=cm

HZ=100

TZ=EST5EDT

TERM=xterms

USER=cm

LANG=C

SHELL=/bin/tcsh

REMOTEHOST=lpscm.gsfc.nasa.gov

REMOTEUSER=UNKNOWN

MAIL=/usr/mail/cm

DISPLAY=198.119.37.190:0

SHLVL=1

PWD=/u04/cm

HOST=lpsdev1

HOSTTYPE=iris4d

MSGVERB=text:action

NOMSGLABEL=1

NOMSGSEVERITY=1

LD_LIBRARYN32_PATH=/usr/lib32

LD_LIBRARY64_PATH=/usr/lib64

LPS_HOME=/u04/cm/lpswork

INIT_HOME=/u04/cm

CM_HOME=/u04/cm/lpswork

PATH=.:/u04/cm/bin:/usr/pvcs/.:/u04/cm/lpswork/ui/bin:/u04/cm/lpswork/RDPS/bin:/u04/cm/lpswork/RDCS/bin:/u04/cm/lpswork/PCDS/bin:/u04/cm/lpswork/MFPS/bin:/u04/cm/lpswork/MACS/bin:/u04/cm/lpswork/LDTS/bin:/u04/cm/lpswork/IDPS/bin:/u04/cm/lpswork/tools/bin:/usr/pure/purif

y/:/u04/cm/lpswork/bin/:/usr/app/oracle/product/7.3.2/bin:/usr/bin/X11/:/usr/local/bin:/usr/bin:/bin:/usr/etc:/etc:/usr/bsd:/usr/sbin:/sbin:/u03/LPS/b3/COTS/hdf/hdfeos/bin/sgi

MANPATH=/usr/pure/purify/man:/u04/cm/lpswork/man:/usr/share/catman:/usr/share/man:/usr/catman:/usr/local/man

LD_LIBRARY_PATH=/usr/app/oracle/product/7.3.2/lib:/usr/lib

X11HOME=/usr/bin/X11

XNLSPATH=/usr/motif/lib/X11/nls

ORACLE_HOME=/usr/app/oracle/product/7.3.2

ORACLE_PATH=/usr/app/oracle/product/7.3.2/bin:/usr/bin/X11/:/usr/local/bin:/usr/bin:/bin:/usr/etc:/etc:/usr/bsd:/usr/sbin:/sbin

ORACLE_SID=LPS

ORACLE_TERM=xterm

TNS_ADMIN=/usr/app/oracle/client/developer2000/1.3.1/tns

TWO_TASK=lps

LPS_BIN=/u04/cm/lpswork/bin

LPS_DANFILE_PATH=/u04/cm/lpswork/DAN

LPS_DDNFILE_PATH=/u04/cm/lpswork/DDN

LPS_JOURNAL_PATH=/u04/cm/lpswork

LPS_OUTFILE_PATH=/u04/cm/lpswork/outfile

LPS_RAWFILE_PATH=/u04/cm/lpswork/rawfile

LPS_REPORT_PATH=/u04/cm/lpswork/reports

LPS_TAPE_DEV=/dev/rmt/tps131d5

LPS_TABLE_PATH=/u04/cm/lpswork/tables

LPS_TEMPFILE_PATH=/u04/cm/lpswork/tmp

LPS_TROUBLEFILE_PATH=/u04/cm/lpswork/troublefile

LPS_IAS_PARMS_PATH=/u04/cm/lpswork/iasparms

LPS_CONT_SCHED_PATH=/u04/cm/lpswork/schedules

LPS_PRINTER_DEVICE=/dev/plp

LPS_TAPE_LIBRARY_DEV=/dev/scsi/sc131d510

RDC_DEVICE=/dev/hpdiB

RDC_STATUS_INTERVAL=30

RDC_THRESH_SYSTEMDISK=0.01

LPS_CAPTURE_PROCESSOR=1

PURIFYHOME=/usr/pure/purify

HDF_HOME=/u04/cm/lpswork/COTS/hdf/4.0r2_IRIX_5.3

HDF_BIN=/u04/cm/lpswork/COTS/hdf/4.0r2_IRIX_5.3/bin

HDF_INC=/u04/cm/lpswork/COTS/hdf/4.0r2_IRIX_5.3/include

HDF_OBJ=/u04/cm/lpswork/COTS/hdf/4.0r2_IRIX_5.3/lib

HDF_SRC=/u04/cm/lpswork/COTS/hdf/4.0r2_IRIX_5.3/src

HDF_SCRIPTS=/u04/cm/lpswork/COTS/hdf/4.0r2_IRIX_5.3/scripts

HDF_EOS=/u04/cm/lpswork/COTS/hdf/hdfeos

HDFEOS_HOME=/u03/LPS/b3/COTS/hdf/hdfeos

HDFLIB=/u03/LPS/b3/COTS/hdf/4.0r2_IRIX_5.3/lib

HDFINC=/u03/LPS/b3/COTS/hdf/4.0r2_IRIX_5.3/include

MACHINE=SGI

OSTYPE=IRIX64

BRAND=sgi

NSL_FLAG=

CC=cc -32

F77=f77 -32

CFLAGS=-O -DIP19 -ansiposix

C_CFH=

CFHFLAGS=-O -DIP19 -ansiposix

C_F77_CFH=

C_F77_LIB=-lI77 -lU77 -lF77

F77FLAGS=

F77_CFH=

F77_C_CFH=

CFH_F77=

F77_C_LIB=

HDFSYS=IRIS4

HDFEOS_INC=/u03/LPS/b3/COTS/hdf/hdfeos/include

HDFEOS_BIN=/u03/LPS/b3/COTS/hdf/hdfeos/bin/sgi

HDFEOS_LIB=/u03/LPS/b3/COTS/hdf/hdfeos/lib/sgi

HDFEOS_OBJ=/u03/LPS/b3/COTS/hdf/hdfeos/obj/sgi

HDFEOS_SRC=/u03/LPS/b3/COTS/hdf/hdfeos/src

COTS_HOME=/u04/cm/lpswork/COTS

DB_HOME=/u04/cm/lpswork/db

DB_INC=/u04/cm/lpswork/db/include

DB_OBJ=/u04/cm/lpswork/db/obj

DB_SRC=/u04/cm/lpswork/db/src

FS_HOME=/u04/cm/lpswork/COTS/frame_sync

FS_INC=/u04/cm/lpswork/COTS/frame_sync/include

FS_OBJ=/u04/cm/lpswork/COTS/frame_sync/obj

FS_SCRIPTS=/u04/cm/lpswork/COTS/frame_sync/scripts

FS_SRC=/u04/cm/lpswork/COTS/frame_sync/src

GLOBAL_HOME=/u04/cm/lpswork/global

GLOBAL_INC=/u04/cm/lpswork/global/include

GLOBAL_OBJ=/u04/cm/lpswork/global/obj

GLOBAL_SRC=/u04/cm/lpswork/global/src

LPS_DATA=/u04/cm/lpswork/data

LPS_SCRIPTS=/u04/cm/lpswork/scripts

TOOLS_HOME=/u04/cm/lpswork/tools

TOOLS_BIN=/u04/cm/lpswork/tools/bin

TOOLS_INC=/u04/cm/lpswork/tools/include

TOOLS_OBJ=/u04/cm/lpswork/tools/obj

TOOLS_SRC=/u04/cm/lpswork/tools/src

UI_HOME=/u04/cm/lpswork/ui

UI_BIN=/u04/cm/lpswork/ui/bin

UI_INC=/u04/cm/lpswork/ui/include

UI_OBJ=/u04/cm/lpswork/ui/obj

UI_SRC=/u04/cm/lpswork/ui/src

IDPS_HOME=/u04/cm/lpswork/IDPS

IDPS_BIN=/u04/cm/lpswork/IDPS/bin

IDPS_DATA=/u04/cm/lpswork/IDPS/data

IDPS_DB=/u04/cm/lpswork/IDPS/db

IDPS_DB_INC=/u04/cm/lpswork/IDPS/db/include

IDPS_DB_OBJ=/u04/cm/lpswork/IDPS/db/obj

IDPS_DB_SRC=/u04/cm/lpswork/IDPS/db/src

IDPS_GLOBAL=/u04/cm/lpswork/IDPS/global

IDPS_GLOBAL_INC=/u04/cm/lpswork/IDPS/global/include

IDPS_GLOBAL_OBJ=/u04/cm/lpswork/IDPS/global/obj

IDPS_GLOBAL_SRC=/u04/cm/lpswork/IDPS/global/src

IDPS_INC=/u04/cm/lpswork/IDPS/include

IDPS_OBJ=/u04/cm/lpswork/IDPS/obj

IDPS_SCRIPTS=/u04/cm/lpswork/IDPS/scripts

IDPS_SRC=/u04/cm/lpswork/IDPS/src

LDTS_HOME=/u04/cm/lpswork/LDTS

LDTS_BIN=/u04/cm/lpswork/LDTS/bin

LDTS_DATA=/u04/cm/lpswork/LDTS/data

LDTS_DB=/u04/cm/lpswork/LDTS/db

LDTS_DB_INC=/u04/cm/lpswork/LDTS/db/include

LDTS_DB_OBJ=/u04/cm/lpswork/LDTS/db/obj

LDTS_DB_SRC=/u04/cm/lpswork/LDTS/db/src

LDTS_GLOBAL=/u04/cm/lpswork/LDTS/global

LDTS_GLOBAL_INC=/u04/cm/lpswork/LDTS/global/include

LDTS_GLOBAL_OBJ=/u04/cm/lpswork/LDTS/global/obj

LDTS_GLOBAL_SRC=/u04/cm/lpswork/LDTS/global/src

LDTS_INC=/u04/cm/lpswork/LDTS/include

LDTS_OBJ=/u04/cm/lpswork/LDTS/obj

LDTS_SCRIPTS=/u04/cm/lpswork/LDTS/scripts

LDTS_SRC=/u04/cm/lpswork/LDTS/src

MACS_HOME=/u04/cm/lpswork/MACS

MACS_BIN=/u04/cm/lpswork/MACS/bin

MACS_DATA=/u04/cm/lpswork/MACS/data

MACS_DB=/u04/cm/lpswork/MACS/db

MACS_DB_INC=/u04/cm/lpswork/MACS/db/include

MACS_DB_OBJ=/u04/cm/lpswork/MACS/db/obj

MACS_DB_SRC=/u04/cm/lpswork/MACS/db/src

MACS_GLOBAL=/u04/cm/lpswork/MACS/global

MACS_GLOBAL_INC=/u04/cm/lpswork/MACS/global/include

MACS_GLOBAL_OBJ=/u04/cm/lpswork/MACS/global/obj

MACS_GLOBAL_SRC=/u04/cm/lpswork/MACS/global/src

MACS_INC=/u04/cm/lpswork/MACS/include

MACS_OBJ=/u04/cm/lpswork/MACS/obj

MACS_SCRIPTS=/u04/cm/lpswork/MACS/scripts

MACS_SRC=/u04/cm/lpswork/MACS/src

MFPS_HOME=/u04/cm/lpswork/MFPS

MFPS_BIN=/u04/cm/lpswork/MFPS/bin

MFPS_DATA=/u04/cm/lpswork/MFPS/data

MFPS_DB=/u04/cm/lpswork/MFPS/db

MFPS_DB_INC=/u04/cm/lpswork/MFPS/db/include

MFPS_DB_OBJ=/u04/cm/lpswork/MFPS/db/obj

MFPS_DB_SRC=/u04/cm/lpswork/MFPS/db/src

MFPS_GLOBAL=/u04/cm/lpswork/MFPS/global

MFPS_GLOBAL_INC=/u04/cm/lpswork/MFPS/global/include

MFPS_GLOBAL_OBJ=/u04/cm/lpswork/MFPS/global/obj

MFPS_GLOBAL_SRC=/u04/cm/lpswork/MFPS/global/src

MFPS_INC=/u04/cm/lpswork/MFPS/include

MFPS_OBJ=/u04/cm/lpswork/MFPS/obj

MFPS_SCRIPTS=/u04/cm/lpswork/MFPS/scripts

MFPS_SRC=/u04/cm/lpswork/MFPS/src

PCDS_HOME=/u04/cm/lpswork/PCDS

PCDS_BIN=/u04/cm/lpswork/PCDS/bin

PCDS_DATA=/u04/cm/lpswork/PCDS/data

PCDS_DB=/u04/cm/lpswork/PCDS/db

PCDS_DB_INC=/u04/cm/lpswork/PCDS/db/include

PCDS_DB_OBJ=/u04/cm/lpswork/PCDS/db/obj

PCDS_DB_SRC=/u04/cm/lpswork/PCDS/db/src

PCDS_GLOBAL=/u04/cm/lpswork/PCDS/global

PCDS_GLOBAL_INC=/u04/cm/lpswork/PCDS/global/include

PCDS_GLOBAL_OBJ=/u04/cm/lpswork/PCDS/global/obj

PCDS_GLOBAL_SRC=/u04/cm/lpswork/PCDS/global/src

PCDS_INC=/u04/cm/lpswork/PCDS/include

PCDS_OBJ=/u04/cm/lpswork/PCDS/obj

PCDS_SCRIPTS=/u04/cm/lpswork/PCDS/scripts

PCDS_SRC=/u04/cm/lpswork/PCDS/src

RDCS_HOME=/u04/cm/lpswork/RDCS

RDCS_BIN=/u04/cm/lpswork/RDCS/bin

RDCS_DATA=/u04/cm/lpswork/RDCS/data

RDCS_DB=/u04/cm/lpswork/RDCS/db

RDCS_DB_INC=/u04/cm/lpswork/RDCS/db/include

RDCS_DB_OBJ=/u04/cm/lpswork/RDCS/db/obj

RDCS_DB_SRC=/u04/cm/lpswork/RDCS/db/src

RDCS_GLOBAL=/u04/cm/lpswork/RDCS/global

RDCS_GLOBAL_INC=/u04/cm/lpswork/RDCS/global/include

RDCS_GLOBAL_OBJ=/u04/cm/lpswork/RDCS/global/obj

RDCS_GLOBAL_SRC=/u04/cm/lpswork/RDCS/global/src

RDCS_INC=/u04/cm/lpswork/RDCS/include

RDCS_OBJ=/u04/cm/lpswork/RDCS/obj

RDCS_SCRIPTS=/u04/cm/lpswork/RDCS/scripts

RDCS_SRC=/u04/cm/lpswork/RDCS/src

RDPS_HOME=/u04/cm/lpswork/RDPS

RDPS_BIN=/u04/cm/lpswork/RDPS/bin

RDPS_DATA=/u04/cm/lpswork/RDPS/data

RDPS_DB=/u04/cm/lpswork/RDPS/db

RDPS_DB_INC=/u04/cm/lpswork/RDPS/db/include

RDPS_DB_OBJ=/u04/cm/lpswork/RDPS/db/obj

RDPS_DB_SRC=/u04/cm/lpswork/RDPS/db/src

RDPS_GLOBAL=/u04/cm/lpswork/RDPS/global

RDPS_GLOBAL_INC=/u04/cm/lpswork/RDPS/global/include

RDPS_GLOBAL_OBJ=/u04/cm/lpswork/RDPS/global/obj

RDPS_GLOBAL_SRC=/u04/cm/lpswork/RDPS/global/src

RDPS_INC=/u04/cm/lpswork/RDPS/include

RDPS_OBJ=/u04/cm/lpswork/RDPS/obj

RDPS_SCRIPTS=/u04/cm/lpswork/RDPS/scripts

RDPS_SRC=/u04/cm/lpswork/RDPS/src

CVINSTRLIB=/u04/cm

PVCSHOME=/usr/pvcs

PVCS_LPS=/usr/pvcs/LPS

CMHOME=/u04/cm/lpswork

DELIV=/u04/cm/Dev_Deliveries/b3.2del/b3

LOGS=/u04/cm/LOGS

ENV=/u04/cm/.kshrc

References

1. Computer Sciences Corporation, 514-3SUG/0195, *Landsat 7 Processing System (LPS) Users Guide*, Release 2, July 1997

2. —, 514-4BIP/0195, *Landsat 7 Processing System (LPS) Build Implementation Plan*, Revision 2, September 1996

3. —, 560-8SDS/0194, *Landsat 7 Processing System (LPS) Detailed Design Specification*, May 26, 1995

4. —, 514-30mm/0196, *Landsat 7 Processing System (LPS) Operations and Maintenance Manual*, Signature Copy, November 27, 1996

Acronyms

CM	configuration management
COTS	commercial off-the-shelf
FTP	File Transfer Protocol
GOTS	Government off-the-shelf
HDF	hierarchical data format
IAS	Image Assessment System
IDPS	image data processing subsystem
IT	integration test
LDTS	LPS data transfer subsystem
LGS	Landsat Ground Station
LPS	Landsat 7 Processing System
MFPS	major frame processing subsystem
PCDS	payload correction data subsystem
PVCS	Polytron Version Control System
RAID	Redundant Array of Inexpensive Devices
RDCS	raw data capture subsystem
TAR	tape archival retrieval